Dalla predizione deterministica all'inferenza variazionale: Var-T-JEPA, LLM-JEPA e le fondamenta probabilistiche delle architetture Joint-Embedding Predictive

Estensioni teoriche, empiriche e cognitive del paradigma JEPA verso l'intelligenza artificiale generale

Abstract

Questo paper analizza tre sviluppi tecnici della famiglia JEPA (Joint Embedding Predictive Architecture) e le loro implicazioni teoriche, empiriche e cognitive. Il primo contributo è la reinterpretazione probabilistica dell'architettura JEPA come modello a variabili latenti accoppiate (Var-JEPA, Gögl e Yau 2026): ogni componente del sistema deterministico canonico emerge come caso limite di un framework bayesiano più ricco, la cui funzione obiettivo è un ELBO a cinque termini che sostituisce strutturalmente lo stop-gradient e l'EMA con meccanismi di prevenzione del collasso fondati sulla teoria variazionale. Il secondo è l'istanziazione di questo framework sul dominio tabulare (Var-T-JEPA), con una strategia di mascheratura a livello di attributo, tokenizzazione eterogenea e quantificazione dell'incertezza latente per selezione selettiva dei campioni. Il terzo è l'estensione degli obiettivi JEPA ai modelli linguistici di grande scala (LLM-JEPA, Huang, LeCun e Balestriero 2025), attraverso una loss composita con predittore a pesi condivisi, maschera di attenzione a blocchi causali e loss dropout come leva computazionale per il preaddestramento a piena scala. I capitoli conclusivi confrontano questi tre framework con l'architettura AMI di LeCun, esaminano il problema dell'unificazione verso l'AGI e identificano i limiti strutturali condivisi che nessuno dei tre contributi risolve.

Introduzione

La presente monografia nasce da una constatazione che attraversa l'intera letteratura recente sull'apprendimento delle rappresentazioni: i sistemi di intelligenza artificiale più potenti che l'umanità abbia mai prodotto sembrano sapere moltissimo e capire pochissimo. Non si tratta di un'iperbole retorica. I grandi modelli linguistici contemporanei generano testi di straordinaria coerenza su qualsiasi argomento; i modelli visivi classificano oggetti con precisione superiore a quella umana in condizioni controllate; i sistemi multimodali descrivono scene, rispondono a domande, producono codice funzionante. Eppure questi stessi sistemi falliscono in modo bizzarro e sistematico di fronte a variazioni minime dei compiti, a riformulazioni innocue delle domande, a combinazioni di concetti mai incontrate nell'addestramento. Questa coesistenza di straordinaria capacità e strutturale fragilità non è un difetto contingente correggibile con più dati o più parametri: è il sintomo di un'inadeguatezza profonda nell'obiettivo stesso dell'apprendimento. Ed è precisamente questa diagnosi che ha motivato lo sviluppo delle architetture JEPA e, in particolare, dei tre contributi che questa monografia analizza.

Il paradigma dominante dell'apprendimento automatico degli ultimi quindici anni — quello che potremmo chiamare paradigma generativo-predittivo a livello sensoriale — ha come obiettivo la ricostruzione o la predizione dell'input grezzo: predire il prossimo token di testo, ricostruire una regione mascherata di un'immagine, generare un frame video coerente con i precedenti. Tale obiettivo ha prodotto risultati empirici di enorme valore pratico. Ma contiene una distorsione strutturale: costringendo il modello a operare nello spazio delle osservazioni sensoriali — pixel, campioni audio, sequenze di token discreti — lo si obbliga a dedicare capacità rappresentativa alla modellazione di varianti percettive semanticamente irrilevanti (il preciso pattern di rumore in un'immagine, le fluttuazioni di fase di un segnale audio, la microtexture di una superficie) a scapito delle strutture causali, relazionali e composizionali che costituiscono il nucleo della comprensione del mondo. La Joint Embedding Predictive Architecture (JEPA), proposta da Yann LeCun e dai suoi collaboratori a partire dal 2022, risponde a questa diagnosi spostando il bersaglio dell'apprendimento: invece di predire osservazioni grezze, il sistema apprende a predire rappresentazioni latenti. La differenza, apparentemente tecnica, ha conseguenze teoriche e pratiche di ampia portata — alcune delle quali questa monografia si propone di esaminare nella loro interezza.

La presente monografia si inserisce nel contesto di un programma di ricerca più ampio, del quale una trattazione epistemologica, storico-genealogica e filosofico-critica è stata sviluppata nella monografia gemella "Rappresentazione, previsione e mondo interno: un'indagine su JEPA e World Models sulla strada verso l'Intelligenza Artificiale Generale" (d'ora in poi Rapp-JEPA). Mentre quella monografia affronta JEPA nella sua interezza — dall'architettura I-JEPA al framework AMI di LeCun, dai World Models di Schmidhuber e Hafner alle estensioni teoriche di LeJEPA e Semantic Tube Prediction, fino ai teoremi di impossibilità di Putnam, Kripke e Tarski-Gödel, e al problema del lavaggio epistemico come meccanismo di sopravvalutazione sistematica dei progressi — il presente testo si concentra su tre sviluppi tecnici specifici e sulle loro implicazioni

teoriche, empiriche e cognitive. Il lettore che desideri un'introduzione all'architettura JEPA nel suo contesto storico e alle questioni filosofiche che essa solleva è invitato a consultare Rapp-JEPA, in particolare i Capitoli I–III e il Capitolo X; per i teoremi di impossibilità richiamati nei Capitoli 19–20 della presente monografia, il riferimento è il Capitolo XI di Rapp-JEPA.

I tre sviluppi tecnici che costituiscono l'oggetto di questo lavoro sono i seguenti. Il primo è la reinterpretazione probabilistica dell'architettura JEPA come modello a variabili latenti accoppiate, proposta da Maxime Gögl e Cian Yau nel loro paper Var-JEPA (2026, arXiv:2603.20111). Questo lavoro risponde a una domanda di "ingegneria inversa probabilistica": quale modello generativo bayesiano avrebbe, come propria procedura naturale di inferenza variazionale ammortizzata, esattamente l'architettura di un sistema JEPA? La risposta che Gögl e Yau articolano con rigore matematico è al tempo stesso ovvia in retrospettiva e inaspettata nelle conseguenze: ogni componente del JEPA deterministico canonico — lo stop-gradient, la media mobile esponenziale del codificatore bersaglio, le penalizzazioni di varianza-covarianza — emerge come caso limite o surrogato approssimato di un meccanismo variazionale principiato. Esplicitare questa struttura latente produce un sistema, Var-JEPA, in cui la prevenzione del collasso rappresentazionale non è affidata a euristiche architetturali ma deriva dall'ottimizzazione di un unico obiettivo variazionale coerente: un ELBO a cinque termini che incorpora ricostruzione del contesto, generazione del bersaglio e tre termini di regolarizzazione KL.

Il secondo sviluppo è l'istanziazione del framework variazionale sul dominio dei dati tabulari, denominata Var-T-JEPA. I dati tabulari costituiscono la forma più ubiqua di informazione strutturata nelle applicazioni reali — diagnosi cliniche, valutazione del rischio creditizio, serie temporali energetiche, profilazione demografica — eppure rimangono storicamente sottorappresentati nella ricerca sull'apprendimento autosupervisionato, che si è sviluppata principalmente attorno a immagini, audio e testo. Il trasferimento di un paradigma architetturale nato per la visione artificiale verso questo dominio non è banale: l'eterogeneità intrinseca degli attributi (variabili numeriche continue, categoriali nominali, ordinali), l'assenza di struttura di vicinanza spaziale o temporale fra le colonne, e la ricchezza di dipendenze non lineari fra attributi impongono revisioni concettuali che vanno ben oltre l'adattamento ingegneristico. Var-T-JEPA risponde a queste sfide con una strategia di mascheratura a livello di attributo, una tokenizzazione eterogenea specifica per tipo di variabile, e — crucialmente — un meccanismo di quantificazione dell'incertezza latente per-campione che nessun approccio puramente deterministico può replicare: la varianza della distribuzione posteriore sul latente bersaglio fornisce un segnale calibrato di affidabilità della rappresentazione, con correlazioni di Spearman rispetto all'incertezza simulata di 0,949 su MNIST corrotto e 0,640 sul dataset sintetico SIM.

Il terzo sviluppo è l'estensione degli obiettivi JEPA ai modelli linguistici di grande scala, proposta da Hu Huang, Yann LeCun e Randall Balestriero nel loro paper LLM-JEPA (2025, arXiv:2509.14252). Questo lavoro colma un'asimmetria ingiustificata nella letteratura: mentre nel dominio visivo gli obiettivi JEPA hanno dimostrato benefici significativi rispetto agli obiettivi di ricostruzione nell'input space, nel dominio del

linguaggio naturale la predizione autoregressiva del token successivo è rimasta pressoché incontrastata come obiettivo di pretraining. LLM-JEPA propone di affiancare — non sostituire — l'obiettivo generativo standard con un termine JEPA che opera nello spazio degli embedding: il modello apprende a predire la rappresentazione latente di una sequenza di codice sorgente a partire dalla rappresentazione latente di una descrizione testuale corrispondente, sfruttando la struttura naturale a due viste di certi dataset (testo→codice, domanda→risposta, problema→soluzione). I risultati empirici su quattro famiglie di modelli (da 1 a 8 miliardi di parametri) e quattro dataset mostrano miglioramenti statisticamente significativi dell'accuratezza — fino a 14 punti percentuali su NL-RX-SYNTH per Llama-3.2-1B — accompagnati da una notevole resistenza all'overfitting e da una struttura quasi-lineare dello spazio latente che spiega meccanicisticamente il beneficio.

La tesi centrale di questa monografia è che questi tre sviluppi non siano contributi isolati ma manifestazioni di un'unica tendenza teorica: la graduale comprensione che l'architettura JEPA canonica è già, implicitamente, un caso particolare di modello a variabili latenti accoppiato, e che esplicitare questa struttura produce benefici tangibili in ogni dominio in cui JEPA viene istanziata. Nel caso di Var-JEPA e Var-T-JEPA, l'esplicitazione produce prevenzione naturale del collasso, incertezza quantificata e unificazione della predizione con la generazione. Nel caso di LLM-JEPA, produce strutturazione geometrica dello spazio latente, riduzione dell'overfitting e convergenza più rapida a parità di budget computazionale. In entrambi i casi, la separazione dichiarata dall'originale formulazione di LeCun dalla modellazione probabilistica generativa si rivela — come Gögl e Yau dimostrano con rigore — più retorica che strutturale: la perdita JEPA è il limite deterministico dell'ELBO variazionale quando le varianze dei posteriori tendono a zero, e i meccanismi euristici anti-collasso sono surrogati imperfetti dei termini di regolarizzazione che il framework variazionale produce naturalmente.

Questa tesi ha un corollario importante per la valutazione del percorso verso l'intelligenza artificiale generale. I capitoli conclusivi della monografia (17–20) confrontano i tre framework con l'architettura AMI di LeCun e con il programma fondazionale di Mumuni e Mumuni (2025) per l'AGI con LLM, mostrando convergenze strutturali e divergenze irriducibili che illuminano tanto i progressi quanto i limiti della ricerca attuale. Il problema del radicamento simbolico, dell'incarnazione, della causalità e della memoria episodica — identificati da Mumuni e Mumuni come i quattro pilastri mancanti verso l'AGI — non vengono risolti da nessuno dei tre contributi tecnici. Ma l'analisi della loro struttura matematica permette di formulare un giudizio più preciso su cosa i framework JEPA variazionali e linguistici possono e non possono fare: non è un pessimismo di principio, ma una distinzione tra progressi nella competenza funzionale e avanzamenti verso la comprensione strutturalmente radicata. Il Capitolo 19, nella sua sezione finale, riprende i tre teoremi di impossibilità discussi in Rapp-JEPA (Putnam, Kripke, Tarski-Gödel) come diagnosi strutturale dei limiti residui, chiarendo perché tali limiti non siano contingenti ma dimostrabili per via logico-matematica.

La monografia si articola in venti capitoli e quattro appendici. I Capitoli 1–4 costruiscono le fondamenta teoriche: il Capitolo 1 diagnostica i limiti strutturali del JEPA canonico attraverso un'analisi geometrica e algebrica del collasso rappresentazionale,

dell'operatore di stop-gradient, della media mobile esponenziale e dei costi ausiliari di regolarizzazione, mostrando che ciascuno di questi meccanismi è un espediente privo di giustificazione principiata; i Capitoli 2–3 costruiscono la reinterpretazione probabilistica, derivando il grafo causale del modello generativo Var-JEPA, la fattorizzazione del posteriore variazionale e l'ELBO a cinque termini; il Capitolo 4 dimostra formalmente la prevenzione del collasso senza stop-gradient né EMA attraverso la geometria del gradiente, la decomposizione ELBO surgery e lo studio sistematico delle varianti di ablazione. I Capitoli 5–6 istanziano il framework sul dominio tabulare: il Capitolo 5 descrive la strategia di mascheratura a livello di attributo, la tokenizzazione eterogenea e i sei moduli della rete variazionale; il Capitolo 6 analizza le proprietà di quantificazione dell'incertezza latente, le curve rischio-copertura, la correlazione con l'incertezza simulata e la connessione con il framework VAE/β-VAE.

I Capitoli 7–11 trattano l'estensione ai modelli linguistici di grande scala: il Capitolo 7 affronta il problema concettuale della definizione delle viste in ambito linguistico e la struttura matematica dell'obiettivo JEPA per LLM; il Capitolo 8 descrive l'architettura della loss composita — il codificatore basato sull'ultimo token, la superiorità della similarità coseno, il predittore a pesi condivisi con token [PRED], la maschera di attenzione a blocchi causali; il Capitolo 9 analizza il loss dropout come meccanismo di riduzione dell'overhead computazionale e la regola empirica di co-aggiustamento di λ e α; i Capitoli 10–11 presentano i risultati empirici su NL-RX-SYNTH, GSM8K, Spider, HellaSwag e NQ-Open su quattro famiglie di modelli, e l'analisi comparativa della resistenza all'overfitting. I Capitoli 12–16 esplorano le implicazioni cognitive e computazionali più ampie: il pre-addestramento con obiettivi JEPA, il radicamento simbolico nei LLM, l'embodiment computazionale, il ragionamento causale e la memoria episodica nei sistemi intelligenti. I Capitoli 17–20, infine, operano il confronto con l'architettura AMI di LeCun, esaminano il problema dell'unificazione verso l'AGI, identificano i limiti strutturali condivisi che nessuno dei tre contributi risolve — dal radicamento simbolico all'apprendimento continuo, dalla costruzione delle viste alla plausibilità fisica delle rappresentazioni — e propongono un programma di ricerca futuro che emerge direttamente dalle lacune dimostrate.

Una nota sul destinatario presupposto e sullo stile. Questo testo è scritto per chi non si accontenta di capire "come funziona" nel senso del ricercatore che vuole replicare un esperimento, né di capire "cosa significa" nel senso del filosofo che non intende sporcarsi le mani con i dettagli tecnici. L'ideale che ha guidato la stesura è quello del lettore che vuole capire perché: perché certe architetture sono più promettenti di altre, perché certi problemi resistono a qualsiasi aumento di scala, perché la distinzione tra competenza funzionale e comprensione genuina non è filosoficamente oziosa ma ha conseguenze ingegneristiche precise. Le derivazioni matematiche — dall'ELBO a cinque termini di Var-JEPA all'analisi spettrale del collasso dimensionale, dalla loss composita di LLM-JEPA alla connessione con il framework β-VAE, dalla decomposizione ELBO surgery alla geometria quasi-lineare dello spazio latente indotto da LLM-JEPA — sono parte integrante della narrazione, non ornamento tecnico separabile. Il testo presuppone familiarità con le basi dell'apprendimento automatico e con i fondamenti del formalismo variazionale (distribuzioni gaussiane, KL-divergenza, ELBO, reparametrization trick);

non presuppone conoscenza della letteratura JEPA, che viene introdotta progressivamente a partire dal Capitolo 1.

Sul piano delle fonti, la letteratura di riferimento principale è quella pubblicata tra il 2020 e il 2026 — un arco di tempo in cui il panorama dell'apprendimento autosupervisionato è mutato con una velocità che rende qualsiasi sintesi parziale per costruzione. Tre lavori costituiscono il nucleo tecnico del testo: il paper Var-JEPA di Gögl e Yau (2026, arXiv:2603.20111), che costruisce la reinterpretazione probabilistica e la istanzia sui dati tabulari; il paper LLM-JEPA di Huang, LeCun e Balestriero (2025, arXiv:2509.14252), che estende gli obiettivi JEPA ai modelli linguistici di grande scala; e la rassegna sull'AGI con LLM di Mumuni e Mumuni (2025, arXiv:2501.03151), che fornisce il framework cognitivo e fondazionale per valutare la portata e i limiti dei contributi tecnici. A questi si aggiunge il lavoro di Thimonier et al. (2025) su T-JEPA deterministico per dati tabulari, che costituisce il punto di riferimento da cui Var-T-JEPA prende le mosse, e il corpus di risultati empirici su I-JEPA (Assran et al., 2023) e V-JEPA (Bardes et al., 2024) che fornisce il contesto applicativo del framework. Sul piano filosofico-critico, i Capitoli 19–20 si appoggiano ai risultati di Putnam, Kripke, Tarski e Gödel, non come citazioni autoritative ma come argomenti formali con conseguenze tecniche dimostrabili: la loro trattazione estesa si trova in Rapp-JEPA, Capitolo XI, ed è qui richiamata in forma sintetica per completare il quadro dei limiti strutturali residui.

Capitolo 1 — I limiti strutturali della predizione deterministica nell'architettura JEPA canonica

1.1 La biforcazione epistemica nel campo dell'apprendimento auto-supervisionato

Il paesaggio dell'apprendimento delle rappresentazioni si è consolidato, nel corso del decennio trascorso, attorno a una frattura concettuale che non è riducibile a una mera preferenza tecnica, ma investe la questione fondamentale di cosa debba significare «comprendere» un segnale. Da un versante si collocano i metodi generativi o ricostruttivi — la tradizione che comprende i modelli autoregressivi a token, i Masked Autoencoders, i modelli a diffusione, e più in generale ogni paradigma che adotti come criterio di addestramento la capacità del sistema di ricostruire fedelmente il proprio ingresso nello spazio delle osservazioni originali. Dal versante opposto si afferma progressivamente un insieme di approcci che rinunciano esplicitamente alla riproduzione del segnale, preferendo invece apprendere rappresentazioni attraverso criteri di coerenza e predicibilità nello spazio latente tra differenti «viste» dello stesso oggetto sottostante.

È in questo secondo filone che si inscrive la Joint Embedding Predictive Architecture, nota con l'acronimo JEPA, formulata da LeCun nel suo manifesto del 2022 come tappa essenziale di un programma più ampio verso sistemi autonomi dotati di un autentico «modello del mondo». La motivazione originale di LeCun è una diagnosi precisa delle insufficienze dei metodi ricostruttivi: costringere un modello a riprodurre pixel, campioni audio o token in tutta la loro specificità equivale a sprecare capacità rappresentazionale per codificare dettagli che nulla aggiungono alla comprensione semantica — il rumore di acquisizione, la texture superficiale, le fluttuazioni stocastiche irrilevanti. Una rappresentazione genuinamente orientata alla percezione e al ragionamento dovrebbe invece catturare le strutture causalmente rilevanti, sopprimendo le variazioni prive di contenuto informativo ai fini della previsione.

Il principio che JEPA incarna è dunque quello della predizione nello spazio delle rappresentazioni: anziché domandare al modello di indovinare il valore grezzo di una porzione mascherata dell'ingresso, gli si chiede di anticipare la rappresentazione codificata di quella porzione, così come verrebbe prodotta da un secondo codificatore denominato «codificatore bersaglio». Questo spostamento ha conseguenze computazionali e teoriche profonde, alcune delle quali favorevoli e altre che costituiscono il tema centrale del presente capitolo. I limiti che si esaminano non sono difetti ingegneristici sormontabili con aggiustamenti marginali: derivano dalla struttura deterministica del paradigma e dalla sua separazione — più retoricamente proclamata che strutturalmente giustificata, come si dimostrerà nei capitoli successivi — dalla modellazione probabilistica generativa.

1.2 Struttura formale dell'architettura JEPA canonica

Un'istanza JEPA nella sua forma più generale opera su coppie di osservazioni $(x, y) \in \mathbb{R}^D \times \mathbb{R}^D$, dove x rappresenta il segnale di contesto e y il segnale bersaglio, tra loro correlati per

costruzione — tipicamente porzioni diverse dello stesso dato grezzo, ottenute per mascheratura o trasformazione. La notazione D denota la dimensione dello spazio delle osservazioni originali, che in contesti visivi può ammontare a decine di migliaia di coordinate e in contesti linguistici è tipicamente sostituita da sequenze di token di lunghezza variabile. Il sistema è composto da tre reti neurali distinte, i cui parametri sono appresi congiuntamente o separatamente a seconda della variante implementativa.

Il codificatore di contesto $f_{ctx} : \mathbb{R}^D \to \mathbb{R}^d$ trasforma l'osservazione di contesto nella sua rappresentazione latente:

$$s_x = f_{ctx}(x) \in \mathbb{R}^d$$

Il codificatore bersaglio $f_{trg} : \mathbb{R}^D \to \mathbb{R}^d$ produce analogamente la rappresentazione del segnale obiettivo:

$$s_y = f_{trg}(y) \in \mathbb{R}^d$$

Il predittore $g : \mathbb{R}^d \times \mathbb{R}^{d_z} \to \mathbb{R}^d$ riceve la rappresentazione di contesto s_x e una variabile ausiliaria $z \in \mathbb{R}^{d_z}$ — che nella formulazione deterministica classica è spesso omessa o fissata a zero, riducendo il predittore a $g : \mathbb{R}^d \to \mathbb{R}^d$ — e produce una stima della rappresentazione bersaglio:

$$\hat{s}_y = g(s_x, z) \in \mathbb{R}^d$$

L'obiettivo di addestramento nella sua forma canonica minimizza la distanza quadratica tra la rappresentazione predetta $\hat{s}_y$ e la rappresentazione bersaglio $\tilde{s}_y = \mathrm{sg}(s_y)$, dove $\mathrm{sg}(\cdot)$ indica l'operatore di stop-gradient — ovvero l'identità nella passata in avanti ma un azzeramento dei gradienti nella retropropagazione:

$$\mathcal{L}_{JEPA} = \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\| \hat{s}_y - \mathrm{sg}(s_y) \|_2^2 \right] = \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\| g(f_{ctx}(x), z) - \mathrm{sg}(f_{trg}(y)) \|_2^2 \right]$$

La minimizzazione di questa perdita chiede al predittore di imparare, a partire dalla rappresentazione di contesto, la rappresentazione che il codificatore bersaglio assegnerebbe al segnale obiettivo. La struttura è intuitivamente appetibile: invece di costringere il sistema a ricostruire i valori grezzi di una regione mascherata, si chiede solo di anticiparne l'embedding astratto. Nelle implementazioni concrete come I-JEPA (Assran et al., 2023) per le immagini e V-JEPA (Bardes et al., 2024) per il video, l'architettura è realizzata su trasformatori visivi che processano patch di immagini o tubelets video, e il codificatore bersaglio è aggiornato mediante una media mobile esponenziale dei parametri del codificatore di contesto, piuttosto che per discesa del gradiente diretta.

1.3 Il collasso rappresentazionale: analisi geometrica e algebrica

Il principale ostacolo che l'architettura deterministica deve affrontare è il collasso rappresentazionale, una trappola degenere intrinseca alla struttura della perdita $\mathcal{L}_{JEPA}$. Per

comprenderlo nella sua profondità occorre analizzare la geometria dell'obiettivo di ottimizzazione.

La perdita $\mathcal{L}$_JEPA è una semi-norma nello spazio prodotto delle uscite del predittore e del codificatore bersaglio. In assenza di vincoli ulteriori, essa ammette una famiglia banale di minimi globali: qualunque configurazione in cui $\hat{sy} = sg(sy)$ per ogni coppia (x, y) nel supporto della distribuzione dei dati $\mathcal{D}$ azzera la perdita. Ciò non richiede che le rappresentazioni contengano informazione utile: è sufficiente che predittore e codificatore bersaglio concordino, indipendentemente da cosa concordino.

La soluzione degenere più pericolosa è quella costante. Se $f_ctx(x) = c$ e $f_trg(y) = c$ per ogni ingresso, con $c \in \mathbb{R}^d$ vettore costante, e $g(c, z) = c$, allora $\mathcal{L}$_JEPA $= 0$ senza che alcuna informazione sia stata estratta dai dati. Questa soluzione è un minimo globale della perdita, ed è raggiungibile per discesa del gradiente attraverso percorsi che, in generale, non presentano barriere energetiche significative. Per vederlo formalmente, si consideri la dinamica del gradiente rispetto ai parametri del codificatore di contesto θ_ctx:

$$\partial \mathcal{L}_JEPA / \partial \theta_ctx = \mathbb{E}\ [\ 2 \cdot (\hat{sy} - sy) \cdot \partial \hat{sy} / \partial \theta_ctx\ ] = \mathbb{E}\ [\ 2 \cdot (g(f_ctx(x)) - f_trg(y)) \cdot (\partial g / \partial sx \cdot \partial f_ctx / \partial \theta_ctx)\ ]$$

Se il sistema si trova vicino alla configurazione collassata $f_ctx(x) \approx c$ e $g(c) \approx f_trg(y) \approx c$, il termine $(g(f_ctx(x)) - f_trg(y))$ è piccolo, il gradiente è quasi nullo, e la discesa del gradiente non produce aggiornamenti significativi: il sistema è intrappolato in un pozzo energetico piatto. Questo è distinto da un minimo locale ordinario perché la perdita è realmente prossima al minimo globale, ma le rappresentazioni apprese sono prive di utilità.

Il collasso non è necessariamente totale; è più frequente osservare forme parziali note come collasso dimensionale. In termini spettrali, la matrice di covarianza empirica delle rappresentazioni bersaglio $C_sy = \mathbb{E}[sy\ sy^\top] - \mathbb{E}[sy]\mathbb{E}[sy]^\top \in \mathbb{R}^{d \times d}$ acquisisce un rango effettivo $r \ll d$. Denotando con $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_d \geq 0$ gli autovalori di C_sy, la dimensionalità intrinseca effettiva è convenzionalmente misurata attraverso la participation ratio:

$$PR(C_sy) = (\Sigma_i \lambda_i)^2 / \Sigma_i \lambda_i^2 \in [1, d]$$

Quando $PR \to 1$, le rappresentazioni sono quasi unidimensionali; quando $PR \to d$, la varianza è distribuita uniformemente su tutte le coordinate — condizione prossima all'isotropia. Nel collasso dimensionale, si osserva $PR \ll d$, con la quasi totalità della varianza concentrata nei primi r autovettori. La qualità delle rappresentazioni come segnale per compiti a valle degrada proporzionalmente: se r dimensioni sono attive e $d - r$ sono degeneri, il modello ha perso $d - r$ gradi di libertà che potrebbero altrimenti codificare struttura semantica rilevante.

1.4 Lo stop-gradient: meccanismo, giustificazione e limiti teorici

La soluzione più diffusa per contrastare il collasso nella famiglia JEPA è l'operatore di stop-gradient $sg(\cdot)$, introdotto sistematicamente nell'ambito di BYOL (Grill et al., 2020) e poi

adottato estensivamente. Il suo meccanismo è architetturale anziché probabilistico: si spezza la simmetria tra codificatore di contesto e codificatore bersaglio impedendo che i gradienti fluiscano attraverso il secondo durante la retropropagazione.

Formalmente, l'effetto dello stop-gradient sulla dinamica del gradiente è il seguente. Indicando con $\theta = (\theta_ctx, \theta_trg, \theta_g)$ il vettore completo dei parametri, gli aggiornamenti per discesa del gradiente sono:

$\Delta\theta_ctx \propto -\partial\mathcal{L}_JEPA/\partial\theta_ctx = -\mathbb{E}\ [\ 2\cdot(\hat{s}y - sy)\ \cdot\ (\partial g/\partial sx)\cdot(\partial f_ctx/\partial\theta_ctx)\ ]$ $\Delta\theta_g \propto -\partial\mathcal{L}_JEPA/\partial\theta_g = -\mathbb{E}\ [\ 2\cdot(\hat{s}y - sy)\ \cdot\ (\partial g/\partial\theta_g)\ ]$ $\Delta\theta_trg = 0$ (per effetto di sg)

Il codificatore bersaglio non riceve alcun gradiente dalla perdita $\mathcal{L}_JEPA$, e deve essere aggiornato attraverso meccanismi alternativi. Nelle implementazioni I-JEPA e V-JEPA, questo meccanismo è l'EMA che verrà discusso nella sezione seguente. In alcune varianti più semplici, il codificatore bersaglio è semplicemente tenuto fisso o aggiornato manualmente.

L'intuizione dietro l'efficacia dello stop-gradient è geometrica: in assenza di gradienti nel ramo bersaglio, il sistema non può minimizzare $\mathcal{L}_JEPA$ riducendo simultaneamente $\hat{s}y$ e sy verso la stessa costante, perché sy non risponde al gradiente. Il codificatore di contesto e il predittore sono dunque «costretti» ad inseguire un bersaglio fisso — le uscite del codificatore bersaglio — che funge da «ancora» che impedisce la deriva collassata.

Tuttavia, questa argomentazione soffre di un vizio di forma: non esclude che il codificatore bersaglio stesso collassi verso una rappresentazione degenere attraverso meccanismi che non passano per il gradiente di $\mathcal{L}_JEPA$. In effetti, se il codificatore bersaglio è aggiornato tramite EMA del codificatore di contesto, e quest'ultimo collassa, allora anche il bersaglio seguirà lentamente il collasso. Lo stop-gradient ritarda il collasso ma non lo preclude strutturalmente.

Un difetto più profondo dello stop-gradient è la sua natura di espediente: non corrisponde alla derivazione di alcun obiettivo principiato. Non esiste una funzione di energia $F(\theta_ctx, \theta_g)$ di cui il gradiente con stop-gradient applicato al ramo bersaglio sia il vero gradiente. L'aggiornamento delle due componenti — (θ_ctx, θ_g) per discesa del gradiente e θ_trg per EMA — definisce una dinamica ibrida che non può essere interpretata come la minimizzazione di un singolo obiettivo coerente. Questo non è un difetto puramente estetico: l'assenza di un obiettivo principiato rende impossibile derivare garanzie di convergenza standard, analizzare teoricamente le proprietà delle rappresentazioni apprese, e guidare principiatamente le scelte architetturali.

1.5 La media mobile esponenziale: stabilizzazione empirica senza fondamento teorico

L'Exponential Moving Average (EMA) del codificatore bersaglio rappresenta un secondo meccanismo di stabilizzazione, tipicamente combinato con lo stop-gradient. I parametri θ_trg del codificatore bersaglio non sono ottimizzati direttamente, ma aggiornati come media mobile di quelli del codificatore di contesto con un coefficiente di momento $\tau \in (0, 1)$, tipicamente nell'intervallo $[0.990, 0.999]$:

$\theta_trg \leftarrow \tau \cdot \theta_trg + (1 - \tau) \cdot \theta_ctx$

Il codificatore bersaglio insegue con inerzia controllata l'evoluzione del codificatore di contesto. Il tempo caratteristico di questa dinamica è $\tau/(1-\tau)$ passi di aggiornamento: per $\tau = 0.996$, questo corrisponde a circa 250 passi. Intuitivamente, il bersaglio «vede» una versione mediata nel tempo del codificatore di contesto, che varia più lentamente del codificatore stesso e fornisce un obiettivo di apprendimento stabile.

L'effetto stabilizzante è documentato empiricamente: in assenza di EMA, i sistemi JEPA tendono a oscillazioni violente durante l'addestramento oppure degenerano rapidamente verso soluzioni collassate. Tuttavia, anche questa soluzione è fondamentalmente euristica. L'aggiornamento EMA con coefficiente $(1 - \tau)$ non corrisponde al gradiente di nessuna funzione obiettivo. Formalmente, se si definisce la funzione $L(\theta_ctx, \theta_trg) = \| \theta_ctx - \theta_trg \|^2$, il cui gradiente rispetto a θ_trg è $2(\theta_trg - \theta_ctx)$, si otterrebbe la regola $\theta_trg \leftarrow \theta_trg - \eta \cdot 2(\theta_trg - \theta_ctx)$, ovvero $\theta_trg \leftarrow (1 - 2\eta)\theta_trg + 2\eta \cdot \theta_ctx$, che coincide con l'EMA solo se $1 - 2\eta = \tau$. Ma questa funzione L non ha nulla a che fare con la qualità delle rappresentazioni apprese: è una semplice penalità di vicinanza tra i parametri dei due codificatori.

La scelta ottimale di τ rimane una questione di taratura empirica specifica al dataset e all'architettura, e non è guidata da alcun principio che connetta τ alle proprietà statistiche dei dati o alla struttura del problema. Gögl e Yau (2026) osservano che sia lo stop-gradient sia l'EMA sono forme di «controllo distributivo surrogato»: impongono indirettamente proprietà sulle rappresentazioni apprese — separazione degli attrattori, diversità dimensionale — che un'architettura variazionale otterrebbe naturalmente attraverso i termini di regolarizzazione KL derivati dal principio di massimizzazione dell'evidenza variazionale.

1.6 I costi ausiliari di regolarizzazione e il proliferare delle euristiche

Una terza linea di difesa contro il collasso è stata elaborata introducendo termini ausiliari espliciti nella funzione di perdita, progettati per preservare la diversità delle rappresentazioni indipendentemente dalla struttura del problema.

Drozdov, Shwartz-Ziv e LeCun (2024) propongono penalità sulla varianza e sulla covarianza degli embedding, nella tradizione dei metodi VICReg. Il termine ausiliario $C(\hat{s}y)$ nella perdita aumentata

$$\mathcal{L}_aug = \mathcal{L}_JEPA + \lambda \cdot C(\hat{s}y)$$

penalizza configurazioni in cui le dimensioni dello spazio latente presentano varianza troppo bassa o covarianza non nulla tra coordinate distinte. In termini algebrici, indicando con $E \in \mathbb{R}^{B \times d}$ la matrice degli embedding bersaglio previsti per un minibatch di dimensione B, i termini tipici sono:

Termine di varianza: $C_var(E) = (1/d) \cdot \Sigma_{j=1}^{d} \max(0, \gamma - Var[E_{:,j}])$ dove $Var[E_{:,j}] = (1/B) \Sigma_{n=1}^{B} (E_{n,j} - \bar{E}_j)^2$ con $\bar{E}_j = (1/B) \Sigma_n E_{n,j}$

Termine di covarianza: $C_cov(E) = (1/d(d-1)) \cdot \Sigma_{i \neq j} [Cov(E_{:,i}, E_{:,j})]^2$ dove $Cov(E_{:,i}, E_{:,j}) = (1/B) \Sigma_n (E_{n,i} - \bar{E}_i)(E_{n,j} - \bar{E}_j)$

Il termine di varianza spinge la varianza marginale di ciascuna dimensione a superare la soglia γ; il termine di covarianza scoraggia le correlazioni tra dimensioni distinte, avvicinando la matrice di covarianza all'identità scalata. Queste penalità agiscono direttamente sulle proprietà statistiche del batch di embedding, indipendentemente dalla struttura del task.

Balestriero e LeCun (2025) propongono nella loro LeJEPA un approccio più sofisticato, denominato Sketched Isotropic Gaussian Regularization (SIGReg). L'argomento motivazionale è che la distribuzione aggregata degli embedding ottimale per il sondaggio a valle è la gaussiana isotropa N(0, I_d), e che questa proprietà dovrebbe essere imposta direttamente attraverso un test statistico. SIGReg applica un test di normalità univariato T — tipicamente il test di Epps-Pulley, che si basa sulla funzione caratteristica empirica — a proiezioni degli embedding lungo un insieme A di direzioni casuali di norma unitaria:

SIGReg(A, {e_n}*{n=1}^N) = (1/|A|) · Σ{a ∈ A} T({a⊤ e_n}_{n=1}^N)

dove {e_n} è un batch di N embedding. La misura quantifica, per ciascuna direzione a, la discrepanza tra la distribuzione empirica delle proiezioni {a⊤ e_n}_n e la normale standard N(0, 1). La media su |A| = M direzioni fornisce una stima della deviazione globale dall'isotropia gaussiana. Il test di Epps-Pulley ha la forma esplicita:

T_EP({x_n}*n) = (1/N) Σ{n=1}^N Σ_{m=1}^N exp(−(x_n − x_m)²/2) − (2/N) Σ_{n=1}^N exp(−x_n²/4) + √2

dove la prima somma misura la varianza della distribuzione empirica, la seconda il suo centramento rispetto a N(0,1), e √2 è una costante di normalizzazione. L'obiettivo SIGReg addestrata sul batch {e_n} tende a N(0, I_d) nel senso delle proiezioni unidimensionali, il che per distribuzioni con varianza finita equivale alla convergenza in distribuzione all'isotropia gaussiana.

Questi meccanismi ausiliari sono efficaci nel ridurre il collasso, ma aggravano la distanza del paradigma JEPA da una derivazione principiata. Ogni termine addizionale introduce iperparametri — il peso λ, la soglia γ, il numero di direzioni M, il numero di frequenze nel test di Epps-Pulley — la cui scelta ottimale non è guidata da alcun principio che emerga naturalmente dalla struttura del problema. L'architettura complessiva diventa una sovrapposizione di euristiche parzialmente motivate, senza che esista un'unica funzione obiettivo da cui tutta la struttura discenda.

1.7 L'impossibilità strutturale della quantificazione dell'incertezza

Al di là del problema del collasso, la struttura deterministica di JEPA porta con sé un limite epistemico più profondo: l'impossibilità di quantificare l'incertezza nella previsione latente. Il predittore g produce, per ciascuna coppia (x, z), un singolo vettore ŝy ∈ ℝ^d. Non esiste alcun meccanismo interno per esprimere che, data una certa osservazione di contesto x, la distribuzione sulle possibili rappresentazioni bersaglio compatibili sia più o meno concentrata, né che alcune dimensioni del latente siano determinate con maggiore certezza di altre.

Questa limitazione ha conseguenze pratiche rilevanti. Nella valutazione selettiva — scenario in cui si desidera astenersi dalla previsione per i campioni la cui incertezza è elevata, costruendo curve rischio-copertura (risk-coverage curves) — un sistema deterministico non dispone di alcun segnale interno su cui basare la scelta di astensione. Tutti i campioni sono trattati epistemicamente in modo equivalente, indipendentemente dalla variabilità intrinseca della relazione contesto-bersaglio nel dato specifico.

Il valore pratico della quantificazione dell'incertezza è dimostrato empiricamente da Gögl e Yau (2026) nel contesto di Var-T-JEPA: il loro sistema, essendo variazionale, produce stime della varianza posteriore per ciascun campione che fungono da proxy dell'incertezza. Sulla versione tabulare di MNIST con corruzioni controllate, la correlazione di Spearman tra l'incertezza latente stimata e il livello di corruzione simulato raggiunge $\varrho = 0.949$; sul dataset sintetico SIM, $\varrho = 0.640$. Le curve rischio-copertura mostrano che scartando il 10%, il 20% o il 50% dei campioni più incerti si ottengono guadagni di accuratezza sistematici e crescenti su tutti i predicatori a valle esaminati. Un sistema JEPA deterministico non è in grado di riprodurre questo comportamento senza l'aggiunta di moduli di stima dell'incertezza post-hoc — che però, non essendo derivati dall'architettura, non hanno garanzie di calibrazione.

La mancanza di quantificazione dell'incertezza è inoltre un ostacolo alla robustezza distribuzionale. Un sistema incapace di riconoscere quando un ingresso si trova lontano dalla distribuzione di addestramento non può segnalare l'inaffidabilità delle proprie rappresentazioni. Questa cecità epistemica è inaccettabile in contesti ad alta responsabilità — diagnosi clinica, guida autonoma, sistemi di raccomandazione critica — e non può essere ovviata senza una revisione fondamentale dell'architettura.

1.8 La dicotomia «predittivo contro generativo» come posizione retorica

Il documento fondativo di LeCun (2022) presenta esplicitamente JEPA come un'alternativa «non generativa» ai modelli a massima verosimiglianza, motivando questa scelta con la volontà di evitare la modellazione del rumore di osservazione. La postura è storicamente comprensibile: i modelli autoregressivi su dati ad alta risoluzione impiegano enormi capacità per restituire dettagli percettivamente irrilevanti. Tuttavia, come Gögl e Yau argomentano con rigore nella loro analisi strutturale, la dicotomia «predittivo contro generativo» è in larga misura retorica.

L'argomentazione è la seguente. Un'architettura JEPA si compone di codificatori (funzioni di inferenza che mappano osservazioni in latenti), di un predittore (una funzione che anticipa la rappresentazione bersaglio dalla rappresentazione di contesto), e di un obiettivo di corrispondenza tra rappresentazioni. Ciascuno di questi elementi ha un preciso analogo nella modellazione variazionale: i codificatori corrispondono ai posteriori variazionali ammortizzati, il predittore corrisponde al prior condizionale appreso sulla variabile latente bersaglio, e l'obiettivo di corrispondenza corrisponde al termine KL che penalizza la deviazione del posteriore dal prior condizionale.

L'intera architettura JEPA è dunque interpretabile, formalmente e non metaforicamente, come un autoencoder variazionale accoppiato con un prior condizionale appreso — e questa interpretazione non è accessoria ma strutturalmente necessaria, nel senso che la perdita JEPA emerge come caso limite deterministico dell'ELBO variazionale quando le varianze dei posteriori tendono a zero. La separazione dichiarata da «modelli generativi a massima verosimiglianza» è dunque una separazione da una specifica classe ristretta di modelli — quelli che minimizzano direttamente la log-verosimiglianza nel spazio delle osservazioni — e non dall'intera tradizione della modellazione probabilistica latente. La dimostrazione formale di questa equivalenza costituisce il nucleo del capitolo successivo.

2.11 Connessioni con i framework di inferenza variazionale gerarchica: BIVA, NVAE e VDVAE

Il modello Var-JEPA, con la sua struttura a tre variabili latenti (s_x, z, s_y) collegate da relazioni di dipendenza definite nel DAG causale, si inserisce in una tradizione più ampia di modelli a variabili latenti gerarchiche che ha conosciuto uno sviluppo notevole nel periodo 2020-2026. Comprendere la posizione di Var-JEPA rispetto a questi framework aiuta a identificare le sue specificità e le sue limitazioni rispetto all'avanguardia della modellazione generativa profonda.

Il Bidirectional-Inference VAE (BIVA, Maaløe et al. 2019) introduce un flusso informativo bidirezionale tra i livelli gerarchici del posteriore variazionale: oltre al flusso bottom-up (dall'osservazione ai latenti), un flusso top-down porta informazione dai latenti superiori ai posteriori dei latenti inferiori, producendo una stima più accurata del vero posteriore. Formalmente, in BIVA il posteriore si fattorizza come q_φ(z₁, ..., z_L | x) = q_φ(z_L | x) · Π_{l=1}^{L-1} q_φ(z_l | z_{l+1}, x), dove ciascun q_φ(z_l | z_{l+1}, x) dipende sia dall'osservazione che dal latente al livello superiore. La connessione con Var-JEPA è strutturalmente interessante: la variabile ausiliaria z in Var-JEPA svolge un ruolo analogo al latente al livello superiore di una gerarchia a due livelli, con s_y condizionato a (s_x, z, y) nel modo che s_y[BIVA] sarebbe condizionato a (z_higher, x) in un BIVA a due livelli.

Il NVAE (Vahdat e Kautz, 2020) e il VDVAE (Child, 2021) spingono questa logica gerarchica a decine di livelli di latenti, dimostrando che la modellazione efficiente di distribuzioni di immagini ad alta risoluzione richiede architetture che operino simultaneamente a scale spaziali diverse. La chiave dell'efficienza computazionale in questi sistemi è il gruppo di residual cells con normalization che implementa il posteriore variazionale in modo differenziabile attraverso i livelli. Per Var-JEPA, questo suggerisce una direzione di estensione naturale: sostituire la singola variabile ausiliaria z con una gerarchia di variabili z₁, ..., z_K, dove ciascun z_k cattura la variabilità del target a una diversa granularità semantica. I livelli superiori della gerarchia catturerebbero la struttura globale (es. la categoria semantica del target), mentre i livelli inferiori catturerebbero i dettagli specifici del campione.

Un'ulteriore connessione teorica riguarda il framework NVAE e la sua gestione del posterior collapse, fenomeno in cui alcuni livelli della gerarchia latente vengono ignorati

dall'ottimizzazione (il posteriore collassa verso il prior senza portare informazione utile). L'analisi di Vahdat e Kautz mostra che il posterior collapse è aggravato dalla presenza di connessioni residuali che consentono all'informazione di bypassare i livelli latenti. In Var-JEPA, il meccanismo analogo è il collasso del latente z verso il prior p(z) = N(0, I) indipendentemente da s_x, situazione che si manifesta empiricamente quando α^KL_z è eccessivamente alto rispetto al segnale di supervisione disponibile. L'annealing separato dei coefficienti α, documentato nella Tabella B.5 dell'Appendice B, è precisamente la tecnica usata da Gögl e Yau per prevenire questo collasso nei primi stadi dell'addestramento.

2.12 Identificabilità e disentanglement nelle rappresentazioni latenti di Var-JEPA

Una questione fondamentale nella valutazione di qualunque modello a variabili latenti è quella dell'identificabilità: in che misura le variabili latenti apprese corrispondono a fattori di variazione semanticamente significativi e indipendenti nei dati? Per un modello VAE standard, Khemakhem et al. (2020) dimostrano nel framework iVAE (identifiable VAE) che le rappresentazioni latenti sono identificabili a meno di permutazioni e rescaling lineari se e solo se il modello è condizionato su una variabile osservabile ausiliaria u che varia con i campioni. Per Var-JEPA, la struttura a due viste (x, y) svolge implicitamente il ruolo di questa variabile ausiliaria.

Formalmente, il problema di identificabilità per Var-JEPA può essere formulato come segue. Siano (θ, φ) e (θ', φ') due parametrizzazioni del modello generativo e variazionale tali che le distribuzioni marginali sui dati coincidano: p_{θ}(x, y) = p_{θ'}(x, y) per quasi ogni (x, y) nella distribuzione dei dati. In che misura le rappresentazioni latenti corrispondenti s_x = E_φ[s_x | x] e s_x' = E_{φ'}[s_x | x] sono equivalenti? La risposta, che richiederebbe un'analisi formale analoga a quella di Khemakhem et al., dipende dalla richezza della classe di modelli considerata e dalle proprietà statistiche della distribuzione congiunta p(x, y). In assenza di vincoli aggiuntivi sulla struttura del decoder U^x_θ, le rappresentazioni latenti sono identificabili solo a meno di trasformazioni arbitrarie che preservano la distribuzione marginale sui dati.

Un risultato positivo parziale emerge dalla struttura del prior condizionale appreso p_θ(s_y | s_x, z). Se il prior condizionale implementa una mappa affine s_y = A·s_x + B·z + c con A, B, c matrici/vettori appresi, allora la struttura della relazione lineare tra s_x e s_y fornisce un vincolo aggiuntivo che riduce l'ambiguità identificativa. Gögl e Yau (2026) osservano empiricamente che le varianze posteriori delle tre componenti latenti (s_x, z, s_y) presentano pattern diversi sui dataset tabulari testati, suggerendo che le tre variabili catturano effettivamente aspetti distinti della struttura dei dati. Una formalizzazione rigorosa di questi pattern in termini di identificabilità rimane un problema aperto che il programma di ricerca futuro (Capitolo 20) dovrebbe prioritizzare.

1.9 Sintesi critica: un paradigma potente, costruito su fondamenta instabili

Al termine di questo percorso analitico, la valutazione che emerge sull'architettura JEPA deterministica è al tempo stesso di apprezzamento per la sua efficacia pratica e di critica fondazionale per la sua struttura teorica. Il paradigma è indubitabilmente potente: ha prodotto sistemi di apprendimento delle rappresentazioni di qualità superiore ai corrispondenti metodi ricostruttivi su compiti di percezione, pianificazione e ragionamento. L'idea di spostare il criterio di apprendimento dallo spazio grezzo delle osservazioni a quello delle rappresentazioni astratte è concettualmente feconda e conferisce al sistema una robustezza che i modelli puramente ricostruttivi non possiedono.

Ma questa potenza è acquistata a un costo epistemico rilevante. Il sistema non sa cosa non sa: non dispone di distribuzioni posteriori, non può esprimere incertezza, e non ha un principio unificante da cui le regolarizzazioni discendano naturalmente. Le soluzioni adottate per prevenire il collasso — stop-gradient, EMA, penalità di varianza-covarianza, SIGReg — sono ingegneristicamente sofisticate ma teoricamente ad hoc, accumulate stratificazione per stratificazione senza che esista un'unica prospettiva da cui l'intera architettura emergesse necessariamente.

La revisione variazionale che Gögl e Yau propongono nel 2026 non rigetta il nucleo dell'idea JEPA — la predizione nello spazio latente — ma lo ricolloca all'interno di un quadro probabilistico completo, dal quale il comportamento deterministico emerge come caso limite e dal quale le proprietà desiderate delle rappresentazioni derivano naturalmente, senza il bisogno di meccanismi surrogati. I capitoli che seguono sviluppano questa revisione nelle sue fondamenta matematiche e nelle sue conseguenze applicative.

1.10 La prospettiva rate-distortion: quanto informazione deve contenere una rappresentazione?

Un inquadramento teorico potente per comprendere perché la struttura deterministica di JEPA sia insufficiente proviene dalla teoria rate-distortion di Shannon. La domanda fondamentale della teoria della compressione con perdita è la seguente: dato un segnale x con distribuzione p(x), e fissato un livello massimo di distorsione D --- misurato attraverso una funzione di perdita $\Delta(x, \hat{x})$ tra il segnale originale e la sua ricostruzione --- qual è la quantità minima di informazione R(D) necessaria per descrivere x in modo da soddisfare il vincolo di distorsione? La risposta, fornita dal teorema rate-distortion di Shannon, è $R(D) = \min_{p(\hat{x}|x): E[\Delta(x,\hat{x})] \leq D} I(X; \hat{X})$, dove $I(X; \hat{X})$ è l'informazione mutua tra il segnale e la sua rappresentazione compressa.

Questa prospettiva illumina il difetto strutturale dei modelli generativi a livello sensoriale: addestrare un modello a minimizzare la perdita di ricostruzione $E[\Delta(x, \hat{x})]$ equivale a operare sul punto operativo (R, D) della curva rate-distortion del dataset. Il problema è che per segnali ad alta dimensionalità --- immagini, audio, testo --- la curva rate-distortion è dominata, nelle sue regioni di bassa distorsione, dal contributo di dimensioni che codificano variabilità percettivamente irrilevante: le fluttuazioni statistiche del rumore di acquisizione, le alte frequenze spaziali della texture, i dettagli fonologici sub-fonemici. Un modello addestrato a minimizzare la distorsione deve allocare capacità rappresentativa a queste dimensioni, consumando budget di rate per informazione che non contribuisce alla comprensione semantica.

JEPA opera su un obiettivo rate-distortion nello spazio delle rappresentazioni latenti anziché nello spazio delle osservazioni. Formalmente, la perdita JEPA $\mathcal{L}_JEPA = E[\|\hat{s}_y - s_y\|^2_2]$ corrisponde a minimizzare la distorsione $D_lat = E[\Delta_lat(s_y, \hat{s}_y)]$ nello spazio latente. Se le rappresentazioni s_y sono state apprese in modo da sopprimere le dimensioni semanticamente irrilevanti --- condensando l'informazione utile in un sottospazio di dimensionalità ridotta --- allora la curva rate-distortion latente è molto più favorevole di quella sensoriale: a parità di distorsione, è sufficiente un rate molto più basso. Questo è il vantaggio fondamentale della predizione in spazio latente: sposta il problema di compressione in un dominio dove la struttura semanticamente rilevante è già stata separata dal rumore. La decomposizione ELBO surgery di Hoffman e Johnson (2016) rende questa connessione esplicita: il termine $I_q(x; s_x)$ nell'ELBO è esattamente il rate allocato alla codifica di x in s_x, e la sua regolarizzazione attraverso il termine KL implementa un collo di bottiglia informativo principiato.

Una conseguenza formale di questa analisi riguarda il regime di alta compressione (β-VAE con $\beta > 1$), dove il termine KL domina la ricostruzione. In questo regime, le componenti di s_x che spiegano poco della varianza di x --- ovvero quelle con bassa informazione mutua parziale rispetto a dimensioni specifiche di x --- vengono soppresse, favorendo rappresentazioni disentangled. Gögl e Yau (2026) non esplorano sistematicamente questo regime per Var-JEPA, ma la connessione con il β-VAE discussa nel Capitolo 6 suggerisce che $\alpha^{KL_sx} > 1$ potrebbe produrre rappresentazioni maggiormente disentangled a scapito della fedeltà ricostruttiva --- un trade-off che la curva rate-distortion latente rende esplicito e gestibile.

1.11 Analisi spettrale del collasso dimensionale: autovalori, participation ratio e dinamica dell'addestramento

Il collasso dimensionale descritto nella sezione 1.3 attraverso la participation ratio $PR(C_sy) = (\Sigma_i \lambda_i)^2 / \Sigma_i \lambda_i^2$ merita un'analisi spettrale più approfondita, in particolare per quanto riguarda la sua dinamica durante l'addestramento e la sua connessione con la struttura dei gradienti. Il collasso dimensionale non è un evento discreto che si verifica in un istante identificabile: è un processo continuo di concentrazione degli autovalori della matrice di covarianza degli embedding verso un numero progressivamente inferiore di componenti dominanti. Formalmente, si può monitorare l'evoluzione di $\lambda_k(t)$ --- il k-esimo autovalore di $C_sy(t)$ al passo di addestramento t --- e osservare come i rapporti λ_k/λ_1 per $k > 1$ tendano verso zero in assenza di meccanismi anti-collasso.

Un risultato importante, derivabile dall'analisi di Jing et al. (2021) sul collasso dimensionale nei metodi SSL contrastivi e adattabile al caso JEPA, riguarda la struttura del gradiente nell'intorno delle configurazioni collassate. Se si definisce la matrice Jacobiana $J_ctx = \partial f_ctx(x)/\partial x \in \mathbb{R}^{d\times D}$ e la matrice $J_trg = \partial f_trg(y)/\partial y \in \mathbb{R}^{d\times D}$, il gradiente di $\mathcal{L}_JEPA$ rispetto ai parametri del context encoder θ_ctx contiene un termine proporzionale a $J_ctx^\top \cdot (\hat{s}_y - s_y)$, dove la differenza $(\hat{s}_y - s_y)$ è la "forza di allineamento" che spinge la rappresentazione predetta verso quella

bersaglio. Quando il sistema si trova vicino al collasso, tutte le rappresentazioni convergono verso la stessa costante c, e sia J_ctx sia J_trg diventano quasi-singolari nella direzione ortogonale a c: i gradienti si concentrano in un sottospazio di dimensione uno, amplificando ulteriormente la concentrazione degli autovalori e creando un regime di retroazione positiva verso il collasso.

La struttura della media mobile esponenziale (EMA) del target encoder introduce una dinamica aggiuntiva in questo scenario. Il target encoder $\theta_trg(t) = \tau \cdot \theta_trg(t-1) + (1-\tau) \cdot \theta_ctx(t)$ è un sistema lineare discreto con costante di tempo $\tau/(1-\tau)$. L'analisi della sua stabilità mostra che per $\tau \to 1$ (EMA molto inerte), la dinamica di θ_trg è quasi-stazionaria rispetto a quella di θ_ctx: il target encoder "vede" una versione mediata nel tempo di θ_ctx, attenuando le oscillazioni ad alta frequenza nello spazio dei parametri. Questo effetto stabilizzante è equivalente, dal punto di vista spettrale, a un filtraggio passa-basso della traiettoria dei parametri nel dominio dell'addestramento. Gögl e Yau dimostrano che nel regime variazionale questa stabilizzazione emerge naturalmente dal termine di generazione $\mathcal{L}_gen$, che "ancora" il target posterior all'osservazione y, rendendo la dinamica del target encoder intrinsecamente più stabile senza la necessità di EMA.

Capitolo 2 — Reinterpretazione probabilistica: JEPA come modello a variabili latenti accoppiato

2.1 L'inversione metodologica come punto di partenza

Il contributo teorico fondamentale di Gögl e Yau (2026) non consiste nell'introduzione di un'architettura visibilmente nuova, ma nell'adozione di una prospettiva radicalmente diversa rispetto a quella con cui JEPA è stato concepito e studiato. La domanda da cui prende avvio la loro analisi è un'inversione metodologica: anziché chiedersi come costruire un sistema JEPA efficace dato un task e dei dati, i due autori chiedono quale modello generativo probabilistico avrebbe, come propria procedura naturale di inferenza variazionale ammortizzata, esattamente l'architettura di un sistema JEPA.

Questa domanda di «ingegneria inversa probabilistica» è legittima e ben posta perché la teoria dell'inferenza variazionale fornisce una mappa quasi biunivoca tra classi di modelli generativi e classi di procedure d'inferenza: specificato un modello generativo nella forma di una distribuzione congiunta $p_\theta(x, z)$ su osservazioni e variabili latenti, l'inferenza variazionale genera un posteriore approssimato $q_\varphi(z|x)$ la cui struttura di fattorizzazione è determinata, attraverso un'ipotesi di indipendenza condizionale (tipicamente l'assunzione di campo medio), dalla topologia del grafo generativo. Il problema inverso — risalire dal posteriore al modello — è sottovincolato in generale, ma diventa determinato quando si impongono vincoli sulla struttura del grafo. Nel caso di JEPA, tali vincoli sono già incorporati nell'architettura: due osservazioni distinte ma correlate (contesto e bersaglio), tre variabili latenti (rappresentazione di contesto, variabile ausiliaria, rappresentazione bersaglio), e un

flusso informativo che va dal contesto al bersaglio attraverso il predittore. Questi elementi suggeriscono univocamente una classe di modelli a variabili latenti accoppiati, che Gögl e Yau denominano Var-JEPA.

2.2 Il grafo causale del modello generativo

Il processo generativo che sottende Var-JEPA è descritto da un grafo aciclico diretto (DAG) su cinque variabili: l'osservazione di contesto x ∈ ℝ^D, l'osservazione bersaglio y ∈ ℝ^D, la rappresentazione latente di contesto sx ∈ ℝ^d, la variabile ausiliaria z ∈ ℝ^{dz}, e la rappresentazione latente bersaglio sy ∈ ℝ^d. Il DAG codifica la seguente struttura di dipendenze causali, descritta attraverso le frecce del processo generativo:

p(sx) → sx → x [sx genera x] p(z) → z → sy [z co-genera sy] sx → sy → y [sx genera sy, che genera y]

In forma compatta, il processo generativo si articola in quattro fasi sequential. Prima si campionano le variabili latenti di contesto e ausiliaria dalle rispettive distribuzioni marginali. Poi il latente bersaglio viene generato condizionatamente a entrambe. Infine le osservazioni x e y vengono generate dai rispettivi latenti.

Questa struttura causale merita un'analisi semantica accurata. Il latente di contesto sx è generato indipendentemente da z e condiziona sia l'osservazione di contesto x che il latente bersaglio sy: questo codifica l'ipotesi che il contesto e il bersaglio condividano informazione strutturale mediata da sx. La variabile ausiliaria z è generata indipendentemente da sx e contribuisce alla generazione di sy ma non di x: questo codifica l'ipotesi che sy contenga variabilità non interamente spiegata da sx — per esempio, la specifica realizzazione di una vista mascherata bersaglio data una rappresentazione di contesto fissa. Le osservazioni x e y sono generate condizionatamente ai soli latenti sx e sy rispettivamente, senza dipendenze dirette tra di esse: la correlazione tra contesto e bersaglio è mediata interamente dai latenti.

È istruttivo confrontare questa struttura con quella di un VAE standard (Kingma & Welling, 2013). In un VAE classico, un singolo latente z genera un'osservazione x, e la procedura di inferenza variazionale produce il posteriore $q_\varphi(z \mid x) \approx p_\theta(z \mid x)$. Il modello Var-JEPA è più ricco: tre latenti distinti, due osservazioni, e una struttura di dipendenze che riflette la relazione asimmetrica tra contesto e bersaglio. La connessione con il Coupled VAE di Hao e Shafto (2023) è immediata: tale modello apprende rappresentazioni condivise di coppie correlate — immagini e descrizioni testuali, audio e spettrogrammi — accoppiando due rami VAE attraverso un termine di regolarizzazione condiviso. Var-JEPA specializza quella struttura aggiungendo la variabile ausiliaria z come «ponte» tra il latente di contesto e quello bersaglio, conferendo al predittore — interpretato come prior condizionale appreso $p_\theta(sy \mid sx, z)$ — la capacità di generare distribuzioni su sy piuttosto che singoli punti.

2.3 Parametrizzazione del modello generativo con distribuzioni gaussiane

Il modello generativo è completamente specificato assegnando forme funzionali alle distribuzioni che compaiono nella fattorizzazione congiunta. Gögl e Yau adottano

distribuzioni gaussiane con parametri apprendibili, scelta che rende le divergenze KL calcolabili in forma chiusa e i gradienti derivabili analiticamente tramite il trucco della riparametrizzazione.

Le distribuzioni a priori sulle variabili latenti di contesto e ausiliaria sono gaussiane standard isotrope:

$$p(sx) = \mathcal{N}(sx\,;\,0, I_d)\quad p(z) = \mathcal{N}(z\,;\,0, I_{dz})$$

La scelta del prior standard è canonica nel framework VAE: impone che le rappresentazioni apprese siano distribuite, in media, secondo una gaussiana isotropa nello spazio latente, favorendo la regolarità e la separazione delle regioni semanticamente distinte.

La verosimiglianza dell'osservazione di contesto dato il corrispondente latente è:

$$p_\theta(x \mid sx) = \mathcal{N}(x\,;\,U^x_\theta(sx), \sigma^2_x \cdot I_D)$$

dove $U^x_\theta : \mathbb{R}^{d_{hid}} \to \mathbb{R}^D$ è la rete decodificatrice del contesto con parametri θ, e σ^2_x è una varianza residua che può essere fissata a 1 o appresa come parametro globale scalare. Analogamente, la verosimiglianza dell'osservazione bersaglio dato il latente bersaglio è:

$$p_\theta(y \mid sy) = \mathcal{N}(y\,;\,U^y_\theta(sy), \sigma^2_y \cdot I_D)$$

Il termine più rilevante, che costituisce la specificità del modello rispetto a due VAE indipendenti, è il prior condizionale del latente bersaglio dati contesto e variabile ausiliaria:

$$p_\theta(sy \mid sx, z) = \mathcal{N}(sy\,;\,\mu^{sy}_\theta(sx, z), \Sigma^{sy}_\theta(sx, z))$$

dove $\mu^{sy}_\theta : \mathbb{R}^d \times \mathbb{R}^{dz} \to \mathbb{R}^d$ è una rete che produce la media della distribuzione predittiva su sy, e $\Sigma^{sy}\theta : \mathbb{R}^d \times \mathbb{R}^{dz} \to \mathbb{R}^{d\times d}\{\text{diag}\}$ è una rete che produce la matrice di covarianza diagonale corrispondente, entrambe con parametri θ. Questo prior condizionale è il cuore dell'architettura: è la componente generativa che corrisponde, nella lettura JEPA, alla funzione del predittore g. Quando $\mu^{sy}_\theta(sx, z) = g(sx, z)$ e $\Sigma^{sy}_\theta \to 0$ (covarianza che tende alla matrice zero), si recupera esattamente la predizione deterministica della JEPA classica. Var-JEPA generalizza dunque il predittore JEPA da una funzione deterministica a una distribuzione condizionale completa: il predittore non restituisce un singolo vettore ma l'intera descrizione probabilistica della distribuzione su sy.

La fattorizzazione congiunta del modello generativo è:

$$p_\theta(x, y, sx, z, sy) = p(sx) \cdot p(z) \cdot p_\theta(x \mid sx) \cdot p_\theta(sy \mid sx, z) \cdot p_\theta(y \mid sy)$$

Ciascun fattore ha un'interpretazione semantica trasparente. I prior p(sx) e p(z) esprimono l'assenza di informazione a priori sui latenti. La verosimiglianza $p_\theta(x|sx)$ lega il latente di contesto all'osservazione corrispondente, assicurando che sx catturi l'informazione rilevante di x. Simmetricamente, $p_\theta(y|sy)$ assicura che sy catturi l'informazione necessaria a spiegare y. Il prior condizionale $p_\theta(sy|sx, z)$ formalizza l'ipotesi generativa che il latente bersaglio sia predicibile dalla coppia (sx, z): la media $\mu^{sy}_\theta(sx, z)$ corrisponde alla previsione del

predittore, e la covarianza $\Sigma^{sy}_\theta(sx, z)$ quantifica l'incertezza del modello su sy dato il contesto.

2.4 Costruzione del posteriore variazionale: fattorizzazione e semantica

Il posteriore esatto $p_\theta(sx, z, sy \mid x, y)$ è intrattatabile per via analitica sotto le parametrizzazioni neurali adottate: nessuno dei tre integrali nella marginalizzazione rispetto a (sx, z, sy) è risolubile in forma chiusa. Si adotta quindi il paradigma dell'inferenza variazionale ammortizzata: si introduce una famiglia di distribuzioni $q_\varphi(sx, z, sy \mid x, y)$, parametrizzata da reti con pesi φ, che approssima il posteriore esatto massimizzando il limite inferiore della log-verosimiglianza marginale.

La scelta della fattorizzazione del posteriore variazionale non è arbitraria: deve rispettare i vincoli informativi dell'architettura JEPA e prevenire i cortocircuiti che vanificherebbero il compito predittivo. Gögl e Yau adottano la fattorizzazione:

$$q_\varphi(sx, z, sy \mid x, y) = q_\varphi(sx \mid x) \cdot q_\varphi(z \mid sx) \cdot q_\varphi(sy \mid sx, z, y)$$

La prima componente $q_\varphi(sx \mid x)$ è il posteriore sul latente di contesto, che dipende unicamente dall'osservazione x. La seconda componente $q_\varphi(z \mid sx)$ è il posteriore sulla variabile ausiliaria, che dipende unicamente dalla rappresentazione di contesto già inferita. La terza componente $q_\varphi(sy \mid sx, z, y)$ è il posteriore sul latente bersaglio, che condiziona sia sulle variabili di contesto (sx, z) sia sull'osservazione bersaglio y.

Tutte e tre le componenti sono parametrizzate come gaussiane con media e covarianza dipendenti dall'input:

$$q_\varphi(sx \mid x) = \mathcal{N}(sx\,;\, \mu^{sx}_\varphi(x), \Sigma^{sx}_\varphi(x)) \quad q_\varphi(z \mid sx) = \mathcal{N}(z\,;\, \mu^{z}_\varphi(sx), \Sigma^{z}_\varphi(sx))$$
$$q_\varphi(sy \mid sx, z, y) = \mathcal{N}(sy\,;\, \mu^{sy}_\varphi(sx, z, y), \Sigma^{sy}_\varphi(sx, z, y))$$

dove le reti di inferenza μ^{sx}_φ, Σ^{sx}_φ, μ^{z}_φ, Σ^{z}_φ, μ^{sy}_φ, Σ^{sy}_φ implementano le corrispondenti funzioni con parametri condivisi φ.

2.5 Giustificazione formale delle tre scelte di fattorizzazione

Le tre scelte di fattorizzazione obbediscono a principi distinti che meritano una giustificazione esplicita, perché determinano in modo decisivo le proprietà del sistema addestrato.

Il principio di separazione del contesto governa la prima scelta. La dipendenza $q_\varphi(sx \mid x)$ senza accesso a y garantisce che sx venga inferito senza «contaminazione» bersaglio. Questa non è una restrizione arbitraria: se sx dipendesse da y, la rappresentazione di contesto incorporerebbe informazione bersaglio non disponibile durante l'inferenza a test-time, quando y è ignota. La separazione garantisce che il sistema impari a codificare in sx

unicamente ciò che x contiene, senza sfruttare shortcut informativi disponibili solo durante l'addestramento.

Il principio di non-leakage della variabile ausiliaria governa la seconda scelta. Poiché z condiziona la generazione di sy ma non di x (nel modello generativo), il suo posteriore $q_\varphi(z|sx)$ non dovrebbe dipendere da y. Se z captasse informazione da y durante l'addestramento, il predittore $p_\theta(sy|sx, z)$ imparerebbe a sfruttare questo canale segreto durante la generazione, producendo previsioni ingannevoli: il sistema si comporterebbe bene durante l'addestramento (quando z «spia» y attraverso $q_\varphi(z|sx,y)$) ma male durante l'inferenza (quando z deve essere campionata dal prior p(z) senza accesso a y). La fattorizzazione $q_\varphi(z|sx)$ chiude strutturalmente questo canale.

Il principio di ancoraggio del latente bersaglio governa la terza scelta. Il posteriore su sy condiziona sia su (sx, z) — stabilendo la connessione con il contesto — sia su y — assicurando che sy contenga informazione sull'osservazione bersaglio. Senza il condizionamento su y, la distribuzione su sy coinciderebbe con il prior condizionale $p_\theta(sy|sx, z)$, e il sistema non avrebbe alcun modo di affinare la stima di sy sulla base del bersaglio osservato durante l'addestramento. Il termine di ricostruzione del bersaglio $\mathbb{E}[\log p_\theta(y|sy)]$, che opera attraverso il posteriore $q_\varphi(sy|sx,z,y)$, «tira» sy verso la regione dello spazio latente da cui y è più facilmente generabile.

2.6 JEPA come caso limite deterministico dell'ELBO variazionale

Uno dei risultati matematicamente più illuminanti dell'analisi di Gögl e Yau è la dimostrazione formale che l'architettura JEPA classica emerge come caso limite deterministico del modello Var-JEPA quando le varianze di tutti i posteriori tendono a zero.

Si considerino le varianze dei tre posteriori:

$\Sigma^{sx}_\varphi \to 0$, $\Sigma^z_\varphi \to 0$, $\Sigma^{sy}_\varphi \to 0$

Nel limite, le distribuzioni posteriori degenerano in distribuzioni di Dirac centrate sulle rispettive medie:

$q_\varphi(sx \mid x) \to \delta(sx - \mu^{sx}_\varphi(x))$ $q_\varphi(z \mid sx) \to \delta(z - \mu^z_\varphi(sx))$ $q_\varphi(sy \mid sx, z, y) \to \delta(sy - \mu^{sy}_\varphi(sx, z, y))$

In questo regime, i latenti sono funzioni deterministiche delle osservazioni. Identificando:

$f_{ctx}(x) \equiv \mu^{sx}_\varphi(x)$ [codificatore di contesto deterministico] $f_{trg}(y) \equiv \mu^{sy}_\varphi(sx, z, y)|\{sx=f_{ctx}(x), z=\mu^z_\varphi(sx)\}$ [codificatore bersaglio deterministico] $g(sx, z) \equiv \mu^{sy}_\theta(sx, z)$ [predittore deterministico]

il prior condizionale $p_\theta(sy|sx, z)$ concentra tutta la sua massa su g(sx, z).

Per vedere come la perdita JEPA emerge dall'ELBO, si consideri il termine KL del posteriore bersaglio. Quando sia il posteriore $q_\varphi(sy|sx,z,y)$ sia il prior condizionale $p_\theta(sy|sx,z)$ sono

gaussiani con covarianze che tendono a zero e medie distinte $\mu^{sy}_\varphi(sx,z,y)$ e $\mu^{sy}_\theta(sx,z)$, la KL tra due gaussiane con covarianze $\sigma^2_{post}\cdot I_d$ e $\sigma^2_{prior}\cdot I_d$ vale:

$$KL(q_\varphi \,||\, p_\theta) = (d/2)\cdot\ln(\sigma^2_{prior}/\sigma^2_{post}) + (1/2\sigma^2_{prior})\cdot\|\mu^{sy}_\varphi - \mu^{sy}_\theta\|^2_2 + (d/2)\cdot(\sigma^2_{post}/\sigma^2_{prior} - 1)$$

Nel limite $\sigma^2_{post} \to 0$ con σ^2_{prior} fissata (o entrambe tendenti a zero con rapporto costante $r = \sigma^2_{post}/\sigma^2_{prior} \to 0$), il termine dominante è:

$$KL \to (1/2\sigma^2_{prior})\cdot\|\mu^{sy}_\varphi(sx,z,y) - \mu^{sy}_\theta(sx,z)\|^2_2$$

Con la corrispondenza $\mu^{sy}_\varphi \to f_{trg}(y) = sy$ e $\mu^{sy}_\theta \to g(sx,z) = \hat{sy}$, questo termine diventa proporzionale a:

$$(1/2\sigma^2_{prior})\cdot\|sy - \hat{sy}\|^2_2 = (1/2\sigma^2_{prior})\cdot\|f_{trg}(y) - g(f_{ctx}(x),z)\|^2_2$$

ovvero esattamente la perdita JEPA $\mathcal{L}_{JEPA}$ (a meno della costante $1/(2\sigma^2_{prior})$). Il termine KL del posteriore bersaglio nell'ELBO variazionale converge dunque alla perdita JEPA nel limite deterministico.

Questo risultato chiarisce retroattivamente il ruolo dello stop-gradient. Nel quadro variazionale, il codificatore bersaglio — che corrisponde al posteriore $q_\varphi(sy|sx,z,y)$ — è addestrato principalmente attraverso il termine di generazione $\mathcal{L}_{gen} = -\mathbb{E}[\log p_\theta(y|sy)]$, che «ancora» sy all'osservazione y senza dipendere dalla qualità del predittore. Nel limite deterministico, quando $\mathcal{L}_{gen}$ è soppressa, l'unico gradiente disponibile per il codificatore bersaglio passerebbe attraverso la perdita $\mathcal{L}_{JEPA}$ verso f_{trg}, ma questo causerebbe il collasso (come analizzato nel capitolo precedente). Lo stop-gradient è dunque un surrogato — necessariamente imperfetto — del termine di generazione che manca nel sistema deterministico: sopprime i gradienti nel ramo bersaglio in assenza di un termine di ricostruzione che li renderebbe non-degenerativi.

2.7 La fattorizzazione congiunta e la struttura dell'ELBO (cenni)

Prima di procedere alla derivazione dettagliata dell'ELBO nel capitolo successivo, è utile anticiparne la struttura qualitativa per illuminare le connessioni con i meccanismi JEPA discussi nel capitolo precedente.

Sostituendo le fattorizzazioni del modello generativo e del posteriore variazionale nell'espressione dell'ELBO, si ottiene una decomposizione in cinque termini distinti, ciascuno con un'interpretazione semantica trasparente e una corrispondenza diretta con un componente dell'architettura:

$$ELBO(x, y; \theta, \varphi) = \mathbb{E}_{q_\varphi(sx|x)}[\log p_\theta(x|sx)]$$ [ricostruzione del contesto] + $\mathbb{E}_{q_\varphi(sx,z,sy|x,y)}[\log p_\theta(y|sy)]$ [generazione del bersaglio] − $KL(q_\varphi(sx|x) \,\|\, p(sx))$ [regolarizzazione di sx] − $KL(q_\varphi(z|sx) \,\|\, p(z))$ [regolarizzazione di z] − $KL(q_\varphi(sy|sx,z,y) \,\|\, p_\theta(sy|sx,z))$ [predittore come prior condizionale]

Il primo termine, la ricostruzione del contesto, assicura che il latente sx contenga informazione sufficiente a spiegare l'osservazione x: esso corrisponde al gradiente che «ancora» il codificatore di contesto a x. Il secondo termine, la generazione del bersaglio, svolge il ruolo analogo per sy rispetto a y: è il termine che nella JEPA deterministica è assente e che rende necessario lo stop-gradient. I tre termini KL regolarizzano rispettivamente sx verso il prior standard (penalizzando il collasso e l'eccessiva complessità della rappresentazione di contesto), z verso il prior standard (assicurando che la variabile ausiliaria non diventi degenere), e sy verso il prior condizionale appreso $p_\theta(sy|sx,z)$ — che è esattamente il predittore g nella lettura JEPA.

È questo quinto termine KL che incarna la previsione latente: la sua minimizzazione forza il posteriore su sy (il codificatore bersaglio) e il prior condizionale su sy (il predittore) a concordare sulla distribuzione di sy dato il contesto. La tensione tra questo termine e il secondo (generazione del bersaglio) produce la dinamica di apprendimento che distingue Var-JEPA dal sistema deterministico: il posteriore su sy deve essere simultaneamente coerente con la previsione del predittore (termine KL) e informativo rispetto all'osservazione bersaglio (termine di generazione), e il punto di equilibrio di questa tensione è una distribuzione su sy che cattura sia la struttura predittiva del contesto sia l'informazione specifica del bersaglio.

2.8 Prevenzione del collasso: meccanismo strutturale nell'ELBO

La prevenzione del collasso nel quadro variazionale non è un risultato contingente di scelte architetturali specifiche, ma una proprietà strutturale dell'ELBO. Per vederlo, si consideri cosa accadrebbe se il codificatore di contesto collassasse verso la rappresentazione identicamente nulla: $f_{ctx}(x) \to 0$ per ogni x, ovvero $\mu^{sx}_\varphi(x) \to 0$.

In quel caso, il primo termine dell'ELBO — la ricostruzione del contesto — divergerebbe verso $-\infty$, poiché una rete decodificatrice deterministica $U^x_\theta(0)$ produce un vettore costante indipendente da x, e la log-verosimiglianza $-\|x - U^x_\theta(0)\|^2/(2\sigma^2_x)$ sarebbe pessima per ogni x non uguale alla costante $U^x_\theta(0)$. Questo termine di ricostruzione penalizza il collasso direttamente: il codificatore di contesto è costretto a mantenere rappresentazioni informative di x, altrimenti il decodificatore non riesce a ricostruire x.

Contemporaneamente, il termine KL su sx — $KL(q_\varphi(sx|x) \,\|\, N(0,I))$ — agirebbe in direzione opposta: una distribuzione $q_\varphi(sx|x)$ collassata su $\mu^{sx}_\varphi(x) = 0$ con varianza nulla sarebbe una distribuzione di Dirac nel centro del prior, minimizzando la KL. Il punto di equilibrio tra questi due impulsi opposti — massimizzare la ricostruzione (che richiede latenti informativi) e minimizzare la KL (che preferisce latenti vicini al prior) — è la rappresentazione «ben compressa» che caratterizza i buoni VAE: informativa quanto basta per la ricostruzione, regolare quanto basta per la struttura latente.

Per sy la situazione è più sottile. Il prior condizionale $p_\theta(sy|sx,z)$ è un prior appreso, non fisso, e dunque non esiste un'ancora globale del tipo N(0,I) che prevenga il collasso di sy verso costanti. È il termine di generazione del bersaglio — $\mathbb{E}[\log p_\theta(y|sy)]$ — a svolgere questa funzione: esso penalizza qualunque configurazione di sy che non permetta al

decodificatore U^y_θ di ricostruire y, richiedendo che sy contenga informazione sull'osservazione bersaglio.

Nel loro studio di ablazione sul dataset sintetico SIM, Gögl e Yau (2026) dimostrano questo meccanismo in modo quantitativo. La variante sperimentale (G), in cui i termini di ricostruzione e generazione sono rimossi ($\alpha_{rec} = \alpha_{gen} = 0$), mantenendo solo i termini KL, ottiene un'accuratezza della sonda lineare di 0.571 ± 0.010 — prossima alla casualità per un problema a tre classi. La variante (I), in cui invece sono rimossi tutti i termini KL mantenendo solo ricostruzione e generazione ($\alpha_{KL_sx} = \alpha_{KL_z} = \alpha_{KL_sy} = 0$), ottiene 0.995 ± 0.002 per il latente sx ma produce una distribuzione aggregata altamente non-gaussiana: la KL aggregata verso N(0,I) raggiunge 10.465 ± 2.207. La variante completa (A), l'ELBO integrale, ottiene 0.996 ± 0.002 con una KL aggregata di 0.113 ± 0.037, dimostrando che i termini KL e quelli di ricostruzione sono mutuamente necessari: né l'uno né l'altro, da solo, è sufficiente per ottenere rappresentazioni di qualità.

2.9 La struttura duale inferenza-generazione

Il modello Var-JEPA esibisce una struttura duale particolarmente elegante tra la fase di addestramento e quella di inferenza a test-time. Nella prima, il sistema riceve coppie (x, y) e le processa attraverso i tre posteriori variazionali per produrre le distribuzioni su (sx, z, sy). Il flusso informativo è:

$$x \rightarrow q_\varphi(sx \mid x) \rightarrow sx \rightarrow q_\varphi(z \mid sx) \rightarrow z \rightarrow q_\varphi(sy \mid sx,z,y) \leftarrow y$$

dove la freccia che porta y nella terza rete di inferenza indica che y contribuisce attivamente alla formazione del posteriore su sy attraverso la rete di inferenza $\mu^{sy}_\varphi(sx, z, y)$.

Nella fase di generazione (inferenza a test-time, con y non disponibile), il flusso cambia:

$$x \rightarrow q_\varphi(sx \mid x) \rightarrow sx,\ z \sim p(z) = \mathcal{N}(0, I),\ sy \sim p_\theta(sy \mid sx, z),\ y \sim p_\theta(y \mid sy)$$

La variabile ausiliaria z è ora campionata dal prior standard piuttosto che inferita dalla rete $q_\varphi(z \mid sx)$: questa differenza è il cuore della separazione tra addestramento e inferenza nel framework variazionale. Il sistema campiona molteplici realizzazioni di $z \sim N(0,I)$, ciascuna delle quali genera attraverso $p_\theta(sy \mid sx,z)$ una diversa distribuzione su sy, e attraverso $p_\theta(y \mid sy)$ una diversa distribuzione su y: l'insieme di queste realizzazioni descrive la varietà di osservazioni bersaglio compatibili con il contesto x, catturando la distribuzione di y condizionata a x nel modo seguente:

$$p_\theta(y \mid x) = \iint p_\theta(y \mid sy) \cdot p_\theta(sy \mid sx,z) \cdot q_\varphi(sx \mid x) \cdot p(z)\, dsy\, dz$$

dove l'integrazione su z rappresenta la marginalizzazione sulla variabilità residua non spiegata dal contesto.

Per le applicazioni a valle orientate alla previsione deterministica — classificazione, regressione, clustering — si adottano le medie posteriori come embedding:

$$sx = \mu^{sx}_\varphi(x),\ z = \mu^z_\varphi(sx),\ sw = \mu^{sy}_\varphi(sx, z, w)$$

dove w indica il vettore di attributi completo nell'implementazione tabulare (descritto nel Capitolo V). Le varianze posteriori corrispondenti — Σ^{sx}_φ(x), Σ^z_φ(sx), Σ^{sy}_φ(sx, z, y) — forniscono naturalmente le stime dell'incertezza per campione, senza moduli aggiuntivi. Gögl e Yau aggregano le deviazioni standard del latente bersaglio per ottenere un punteggio scalare di incertezza per campione, che viene poi utilizzato nella valutazione selettiva.

2.10 Collocazione nella letteratura e distinzione dai lavori concorrenti

Il modello Var-JEPA si colloca all'intersezione di due tradizioni di ricerca che hanno proceduto in modo relativamente indipendente. La tradizione JEPA ha prodotto I-JEPA (Assran et al., 2023), V-JEPA (Bardes et al., 2024) e T-JEPA (Thimonier et al., 2025), dimostrando la superiorità dei criteri di predizione nello spazio latente per l'apprendimento di rappresentazioni orientate alla percezione, ma senza fondamento probabilistico esplicito. La tradizione VAE ha sviluppato una teoria probabilistica dell'apprendimento non supervisionato con garanzie di regolarità, incertezza quantificata e obiettivi derivabili da principi, ma senza applicazioni sistematiche alla predizione tra viste.

Il lavoro concorrente di Huang (2026), menzionato dagli stessi Gögl e Yau, propone una formulazione probabilistica di JEPA denominata V-JEPA (da non confondere con il V-JEPA per il video di Bardes et al. 2024), anch'essa orientata alla quantificazione dell'incertezza nella previsione latente. La distinzione fondamentale, sottolineata dagli autori, è che Var-JEPA non è un regolarizzatore incrementale aggiunto sopra l'architettura JEPA, ma una riformulazione generativa completa da cui l'architettura JEPA emergerebbe come caso speciale nel limite deterministico. Huang (2026) formula il problema come stima bayesiana di un'incertezza sulla previsione latente, operando dall'interno della struttura JEPA; Gögl e Yau costruiscono invece un modello generativo completo con un ELBO unificato che incorpora ricostruzione, generazione, regolarizzazione e previsione in un unico principio variazionale.

La novità del contributo non risiede nella singolarità dei componenti architetturali, tutti pre-esistenti nella letteratura, ma nella precisione con cui si dimostra che JEPA è già, implicitamente, un caso particolare di modello a variabili latenti accoppiato, e che esplicitare questa struttura produce benefici tangibili — prevenzione naturale del collasso, incertezza quantificata, unificazione della predizione con la generazione — senza sacrificare le qualità rappresentazionali che hanno reso JEPA attraente. La dimostrazione che la perdita JEPA è il limite deterministico della KL variazionale è il contributo formale che lega i due paradigmi e che motiva la costruzione dell'architettura Var-T-JEPA per dati tabulari, descritta nel Capitolo V.

Capitolo 3 — Derivazione dell'evidenza inferiore variazionale e anatomia dei cinque termini

3.1 Il problema dell'intrattabilità e la strategia variazionale

Il punto di partenza di ogni procedura di inferenza variazionale è il riconoscimento di un'intrattabilità fondamentale. L'obiettivo teorico di Var-JEPA è la massimizzazione della log-verosimiglianza marginale log $p_\theta(x, y)$ rispetto ai parametri generativi θ, dove la marginalizzazione abbraccia i tre spazi latenti distinti del modello:

$$\log p_\theta(x, y) = \log \iiint p_\theta(x, y, sx, z, sy)\, dsx\, dz\, dsy$$

Sostituendo la fattorizzazione congiunta del modello generativo derivata nel capitolo precedente — $p_\theta(x,y,sx,z,sy) = p(sx)\cdot p(z)\cdot p_\theta(x \mid sx)\cdot p_\theta(sy \mid sx,z)\cdot p_\theta(y \mid sy)$ — si ottiene:

$$\log p_\theta(x, y) = \log \iiint p(sx)\cdot p(z)\cdot p_\theta(x \mid sx)\cdot p_\theta(sy \mid sx,z)\cdot p_\theta(y \mid sy)\, dsx\, dz\, dsy$$

Questo integrale triplo non è risolubile in forma chiusa per tre ragioni concorrenti. Le reti generative U^x_θ, U^y_θ e le reti predittive μ^{sy}_θ, Σ^{sy}_θ sono funzioni non-lineari dei loro argomenti, il che impedisce qualunque riduzione analitica delle integrali. Lo spazio latente congiunto (sx, z, sy) ha dimensionalità d + dz + d, tipicamente dell'ordine di alcune centinaia di coordinate, rendendo l'integrazione numerica per quadratura inapplicabile. I percorsi di dipendenza nel grafo aciclico diretto — in particolare la catena sx → sy e z → sy — rendono l'integrale non fattorizzabile: l'integrazione su sy non può essere separata da quella su sx e z, poiché $p_\theta(sy \mid sx,z)$ dipende da entrambi simultaneamente.

La strategia variazionale affronta questa intrattabilità attraverso un'idea elegante: anziché calcolare direttamente log $p_\theta(x,y)$, si massimizza un suo limite inferiore computabile, noto come evidenza inferiore variazionale (ELBO, Evidence Lower BOund). Il limite è ottenuto introducendo una distribuzione ausiliaria $q_\varphi(sx,z,sy \mid x,y)$, denominata posteriore variazionale, e applicando la disuguaglianza di Jensen alla funzione logaritmo.

3.2 Derivazione passo per passo dell'ELBO

La derivazione dell'ELBO procede attraverso un'unica catena di eguaglianze e disuguaglianze. Si parte dalla log-verosimiglianza marginale e si introduce la distribuzione variazionale moltiplicando e dividendo per q_φ all'interno dell'integrale:

$$\log p_\theta(x, y) = \log \iiint p_\theta(x, y, sx, z, sy)\, dsx\, dz\, dsy = \log \iiint \left[p_\theta(x, y, sx, z, sy) \,/\, q_\varphi(sx, z, sy \mid x, y)\right] \cdot q_\varphi(sx, z, sy \mid x, y)\, dsx\, dz\, dsy = \log \mathbb{E}_{q_\varphi(sx,z,sy \mid x,y)}\left[p_\theta(x, y, sx, z, sy) \,/\, q_\varphi(sx, z, sy \mid x, y)\right]$$

L'applicazione della disuguaglianza di Jensen — nella forma $\log \mathbb{E}_q[f] \geq \mathbb{E}_q[\log f]$, valida per qualsiasi funzione f misurabile non-negativa grazie alla concavità del logaritmo — produce:

$$\log p_\theta(x, y) \geq \mathbb{E}_{q_\varphi(sx,z,sy \mid x,y)}\left[\log\left(p_\theta(x, y, sx, z, sy) \,/\, q_\varphi(sx, z, sy \mid x, y)\right)\right] = \mathbb{E}_{q_\varphi}\left[\log p_\theta(x, y, sx, z, sy) - \log q_\varphi(sx, z, sy \mid x, y)\right] \equiv \mathrm{ELBO}(x, y; \theta, \varphi)$$

Il divario tra la log-verosimiglianza e l'ELBO non è arbitrario: per ogni scelta di q_φ, vale l'identità:

$$\log p_\theta(x, y) - \mathrm{ELBO}(x, y; \theta, \varphi) = \mathbb{E}_{q_\varphi}[\log q_\varphi(sx, z, sy \mid x, y) - \log p_\theta(sx, z, sy \mid x, y)] = \mathrm{KL}(q_\varphi(sx, z, sy \mid x, y) \,\|\, p_\theta(sx, z, sy \mid x, y)) \geq 0$$

dove $p_\theta(sx, z, sy \mid x, y) = p_\theta(x, y, sx, z, sy) / p_\theta(x, y)$ è il vero posteriore del modello, che è la distribuzione che q_φ approssima. Poiché la divergenza KL è sempre non-negativa, l'ELBO è davvero un limite inferiore: $\log p_\theta(x,y) \geq \mathrm{ELBO}$ per ogni q_φ, con uguaglianza se e solo se $q_\varphi = p_\theta(\cdot \mid x,y)$. La massimizzazione dell'ELBO rispetto a (θ, φ) è dunque equivalente alla minimizzazione simultanea del gap KL, avvicinando il posteriore variazionale alla vera distribuzione a posteriori del modello.

3.3 Sviluppo dell'ELBO con le fattorizzazioni esplicite

L'espansione analitica dell'ELBO richiede di sostituire le fattorizzazioni del modello generativo e del posteriore variazionale. Si scrive:

$$\log p_\theta(x, y, sx, z, sy) = \log [p(sx) \cdot p(z) \cdot p_\theta(x \mid sx) \cdot p_\theta(sy \mid sx, z) \cdot p_\theta(y \mid sy)] = \log p(sx) + \log p(z) + \log p_\theta(x \mid sx) + \log p_\theta(sy \mid sx, z) + \log p_\theta(y \mid sy)$$

$$\log q_\varphi(sx, z, sy \mid x, y) = \log [q_\varphi(sx \mid x) \cdot q_\varphi(z \mid sx) \cdot q_\varphi(sy \mid sx, z, y)] = \log q_\varphi(sx \mid x) + \log q_\varphi(z \mid sx) + \log q_\varphi(sy \mid sx, z, y)$$

Sostituendo nell'espressione dell'ELBO:

$$\mathrm{ELBO}(x, y; \theta, \varphi) = \mathbb{E}_{q_\varphi}[\log p(sx) + \log p(z) + \log p_\theta(x \mid sx) + \log p_\theta(sy \mid sx,z) + \log p_\theta(y \mid sy) - \log q_\varphi(sx \mid x) - \log q_\varphi(z \mid sx) - \log q_\varphi(sy \mid sx,z,y)]$$

Si raggruppano i termini secondo le variabili latenti coinvolte, separando i contributi che dipendono da sx, quelli che dipendono da z, e quelli che dipendono da sy:

$$\mathrm{ELBO} = \mathbb{E}_{q_\varphi(sx \mid x)}[\log p_\theta(x \mid sx)] \text{ (A)} + \mathbb{E}_{q_\varphi(sx,z,sy \mid x,y)}[\log p_\theta(y \mid sy)] \text{ (B)} + \mathbb{E}_{q_\varphi(sx \mid x)}[\log p(sx) - \log q_\varphi(sx \mid x)] \text{ (C)} + \mathbb{E}_{q_\varphi(sx,z \mid x)}[\log p(z) - \log q_\varphi(z \mid sx)] \text{ (D)} + \mathbb{E}_{q_\varphi(sx,z,sy \mid x,y)}[\log p_\theta(sy \mid sx,z) - \log q_\varphi(sy \mid sx,z,y)] \text{ (E)}$$

Applicando l'identità fondamentale dell'inferenza variazionale — $\mathbb{E}_q[\log p - \log q] = -\mathrm{KL}(q \,\|\, p)$ — valida per qualsiasi coppia di densità (p, q), i termini (C), (D) ed (E) si contraggono ciascuno in una divergenza KL:

$$\text{(C)} = -\mathrm{KL}(q_\varphi(sx \mid x) \,\|\, p(sx)) \quad \text{(D)} = -\mathrm{KL}(q_\varphi(z \mid sx) \,\|\, p(z)) \quad \text{(E)} = -\mathrm{KL}(q_\varphi(sy \mid sx,z,y) \,\|\, p_\theta(sy \mid sx,z))$$

dove il termine (D) richiede una precisazione: poiché $q_\varphi(z \mid sx)$ dipende da sx che è a sua volta campionato da $q_\varphi(sx \mid x)$, l'aspettazione in (D) è presa rispetto alla distribuzione marginale $q_\varphi(sx,z \mid x) = q_\varphi(sx \mid x) \cdot q_\varphi(z \mid sx)$, e la KL è dunque:

$-\mathrm{KL}(q_\varphi(z|sx) \,\|\, p(z)) = -\mathbb{E}_{q_\varphi(sx|x)}\,[\mathrm{KL}(q_\varphi(z|sx) \,\|\, p(z))]$

ovvero la KL mediata sulla distribuzione di sx.

L'ELBO completo è dunque:

$\mathrm{ELBO}(x, y; \theta, \varphi) = \mathbb{E}_{q_\varphi(sx|x)}\,[\log p_\theta(x|sx)]$ [ricostruzione del contesto: $\mathcal{L}_{rec}$ negata]

- $\mathbb{E}_{q_\varphi(sx,z,sy|x,y)}\,[\log p_\theta(y|sy)]$ [generazione del bersaglio: $\mathcal{L}_{gen}$ negata] − $\mathrm{KL}(q_\varphi(sx|x) \,\|\, p(sx))$ [KL su sx: $\mathcal{L}^{KL}_{sx}$] − $\mathbb{E}_{q_\varphi(sx|x)}[\mathrm{KL}(q_\varphi(z|sx) \,\|\, p(z))]$ [KL su z mediata: $\mathcal{L}^{KL}_{z}$] − $\mathbb{E}_{q_\varphi(sx,z|x)}[\mathrm{KL}(q_\varphi(sy|sx,z,y) \,\|\, p_\theta(sy|sx,z))]$ [KL su sy: $\mathcal{L}^{KL}_{sy}$]

Convertendo in problema di minimizzazione — coerente con l'ottimizzazione per discesa del gradiente stocastico — si introduce la funzione di perdita $\mathcal{L}_{ELBO}$ come negativo dell'ELBO, pesato da coefficienti scalari $\alpha \geq 0$:

$\mathcal{L}_{ELBO}(x, y; \theta, \varphi) = \alpha^{rec} \cdot \mathcal{L}_{rec} + \alpha^{gen} \cdot \mathcal{L}_{gen} + \alpha^{KL}_{sx} \cdot \mathcal{L}^{KL}_{sx} + \alpha^{KL}_{z} \cdot \mathcal{L}^{KL}_{z} + \alpha^{KL}_{sy} \cdot \mathcal{L}^{KL}_{sy}$

con le cinque componenti definite esplicitamente:

$\mathcal{L}_{rec} = -\mathbb{E}_{q_\varphi(sx|x)}\,[\log p_\theta(x|sx)]$ $\mathcal{L}_{gen} = -\mathbb{E}_{q_\varphi(sx,z,sy|x,y)}\,[\log p_\theta(y|sy)]$ $\mathcal{L}^{KL}_{sx} = KL(q_\varphi(sx|x) \,\|\, p(sx))$ $\mathcal{L}^{KL}_{z} = \mathbb{E}_{q_\varphi(sx|x)}\,[KL(q_\varphi(z|sx) \,\|\, p(z))]$ $\mathcal{L}^{KL}_{sy} = \mathbb{E}_{q_\varphi(sx,z|x)}\,[\mathrm{KL}(q_\varphi(sy|sx,z,y) \,\|\, p_\theta(sy|sx,z))]$

La massimizzazione dell'ELBO corrisponde alla minimizzazione di $\mathcal{L}_{ELBO}$ con tutti i coefficienti α posti all'unità. I coefficienti scalari servono a modulare il contributo relativo di ciascun termine durante l'addestramento — in particolare per gestire l'annealing del peso KL, tecnica comune nei VAE per prevenire il collasso posteriore durante le fasi iniziali dell'ottimizzazione.

3.4 Il termine di ricostruzione del contesto: $\mathcal{L}_{rec}$

Il primo termine $\mathcal{L}_{rec}$ misura la qualità con cui la rappresentazione latente sx riesce a restituire l'osservazione di contesto x attraverso il decodificatore. Nella parametrizzazione gaussiana adottata — $p_\theta(x|sx) = \mathcal{N}(x; U^x_\theta(sx), \sigma^2_x \cdot I_D)$ — il logaritmo della densità condizionale è:

$$\log p_\theta(x|sx) = -(D/2)\cdot\log(2\pi\sigma^2_x) - (1/(2\sigma^2_x)) \cdot \|x - U^x_\theta(sx)\|^2_2$$

Il termine costante $-(D/2)\cdot\log(2\pi\sigma^2_x)$ non dipende dai parametri θ quando σ^2_x è fissata, e non influisce sui gradienti. La perdita di ricostruzione diviene dunque, sotto stima Monte Carlo con un singolo campione $sx \sim q_\varphi(sx|x)$:

$$\mathcal{L}_{rec} \approx (1/(2\sigma^2_x)) \cdot \|x - U^x_\theta(sx)\|^2_2 + \text{costante}$$

che corrisponde a un errore quadratico medio pesato dall'inverso della varianza residua. Quando σ^2_x è appresa globalmente come parametro scalare, la sua ottimizzazione bilancia

automaticamente il contributo di $\mathcal{L}$_rec nella perdita totale: se σ^2_x aumenta, la perdita di ricostruzione si riduce ma il logaritmo nel termine KL di una gaussiana con varianza σ^2_x (che compare implicitamente quando si calcola l'ELBO completo) aumenta, producendo un trade-off automatico.

La funzione di $\mathcal{L}$_rec è duplice. Dal lato del decodificatore θ: esso addestra la rete U^x_θ a restituire x con fedeltà quanto più elevata, imparando la mappa inversa dal latente all'osservazione. Dal lato del codificatore variazionale φ: esso vincola la rete q_φ(sx|x) a estrarre da x un codice latente sufficientemente ricco da rendere tale ricostruzione possibile. Se il posteriore su sx si concentra su un punto fisso indipendente da x, la log-verosimiglianza $-\mathcal{L}$_rec sarebbe costantemente bassa, indipendentemente dalla struttura dei dati: il termine di ricostruzione impedisce questa degenerazione.

Vale la pena sottolineare che $\mathcal{L}$_rec assolve in Var-JEPA una funzione strutturalmente diversa da quella che svolge in un autoencoder classico: non è il bersaglio dell'apprendimento — il sistema non è addestrato a ricostruire bene x per sé — ma un vincolo che garantisce l'informatività minima del codice latente. In questo senso, $\mathcal{L}$_rec è uno strumento di prevenzione del collasso, non un obiettivo percettivo.

3.5 Il termine generativo del bersaglio: $\mathcal{L}$_gen

Il secondo termine $\mathcal{L}$_gen svolge il ruolo simmetrico di $\mathcal{L}$_rec rispetto al latente bersaglio sy. Nella parametrizzazione gaussiana p_θ(y|sy) = $\mathcal{N}$(y; U^y_θ(sy), σ^2_y·I_D):

$$\log p_\theta(y|sy) = -(D/2)\cdot\log(2\pi\sigma^2_y) - (1/(2\sigma^2_y)) \cdot \|y - U^y_\theta(sy)\|^2_2$$

Il termine $\mathcal{L}$_gen è computato attraverso la catena di campionamento: prima sx ~ q_φ(sx|x), poi z ~ q_φ(z|sx), poi sy ~ q_φ(sy|sx,z,y). Quest'ultimo campionamento richiede il reparametrization trick — sy = μ^{sy}_φ(sx,z,y) + Σ^{sy}_φ(sx,z,y)^{1/2} · ε_sy con ε_sy ~ $\mathcal{N}$(0,I_d) — per rendere il gradiente rispetto a φ calcolabile attraverso campionamento stocastico.

La pressione esercitata da $\mathcal{L}$_gen sul posteriore variazionale q_φ(sy|sx,z,y) è la seguente: esso spinge la media μ^{sy}_φ(sx,z,y) verso valori da cui il decodificatore U^y_θ può ricostruire y con piccolo errore. Detto in termini geometrici: la distribuzione variazionale su sy deve concentrare la sua massa sulla regione dello spazio latente che è «decodificabile in y». Questo è precisamente il ruolo del codificatore bersaglio nella JEPA classica — f_trg(y) deve produrre la rappresentazione «corretta» di y — ma qui realizzato in modo probabilistico e con una funzione obiettivo principiata.

$\mathcal{L}$_gen è il termine assente nella JEPA deterministica classica, e la sua assenza è la causa ultima della necessità dello stop-gradient. Come mostrato nel capitolo precedente, nel limite deterministico il termine $\mathcal{L}$_gen scompare insieme alla varianza posteriore, e la sola pressione che rimane sul codificatore bersaglio è il gradiente della perdita $\mathcal{L}$_JEPA attraverso il ramo bersaglio — che però, se non bloccato dallo stop-gradient, porterebbe al collasso. $\mathcal{L}$_gen è dunque la sostituzione principiata dello stop-gradient nel quadro variazionale.

3.6 Il termine KL sulla rappresentazione di contesto: $\mathcal{L}^{KL}_{sx}$

Il terzo termine regolarizza il posteriore variazionale q_φ(sx|x) verso il prior standard N(0, I_d). Per la parametrizzazione gaussiana diagonale adottata — q_φ(sx^{(j)}|x) = $\mathcal{N}$(sx^{(j)}; μ^{sx}*φ(x)^{(j)}, diag(Σ^{sx}φ(x)^{(j)})) per il j-esimo token di contesto — la KL tra due gaussiane è calcolabile in forma chiusa. Per una gaussiana q = $\mathcal{N}$(μ_q, Σ_q) con Σ_q = diag(σ²{q,1}, ..., σ²{q,d}) e il prior p = $\mathcal{N}$(0, I_d), la formula generale

$$KL(\mathcal{N}(\mu_q, \Sigma_q) \| \mathcal{N}(\mu_p, \Sigma_p)) = \tfrac{1}{2} \cdot [\mathrm{tr}(\Sigma_p^{-1} \Sigma_q) + (\mu_p-\mu_q)^T\Sigma_p^{-1}(\mu_p-\mu_q) + \log(\det \Sigma_p/\det \Sigma_q) - d]$$

si specializza con μ_p = 0 e Σ_p = I_d a:

$$KL(\mathcal{N}(\mu_q, \Sigma_q) \| \mathcal{N}(0, I_d)) = \tfrac{1}{2} \cdot [\mathrm{tr}(\Sigma_q) + \|\mu_q\|^2_2 + \log(1/\det \Sigma_q) - d] = \tfrac{1}{2} \cdot \Sigma_{k=1}^{d} [\sigma^2_{q,k} + \mu^2_{q,k} - \log(\sigma^2_{q,k}) - 1]$$

Per il j-esimo token di contesto, la KL per campione è dunque:

$$KL(q_\varphi(sx^{(j)}|x) \| \mathcal{N}(0,I)) = \tfrac{1}{2} \cdot \Sigma_{k=1}^{D_{hid}} [\Sigma^{(j,k)}\{sx,\varphi\} + (\mu^{(j,k)}\{sx,\varphi\})^2 - \log \Sigma^{(j,k)}_{sx,\varphi} - 1]$$

dove D_hid è la dimensione dell'embedding latente. La perdita $\mathcal{L}^{KL}_{sx}$ è la media di questa quantità su tutti i token di contesto:

$$\mathcal{L}^{KL}\{sx\} = (1/M_{ctx}) \cdot \Sigma\{j \in m_{ctx}\} KL(q_\varphi(sx^{(j)}|x) \| \mathcal{N}(0,I))$$

La funzione del termine $\sigma^2_{q,k} + \mu^2_{q,k} - \log(\sigma^2_{q,k}) - 1$ come funzione di $\sigma^2_{q,k} \geq 0$ ha un unico minimo in $\sigma^2_{q,k} = 1$, dove vale 0. Analogamente, $\mu^2_{q,k}$ ha minimo in $\mu_{q,k} = 0$, dove vale 0. La KL è dunque non-negativa e si annulla se e solo se μ_q = 0 e Σ_q = I_d — ovvero quando il posteriore coincide esattamente con il prior. Questo termine esercita dunque una pressione che spinge le medie dei latenti verso l'origine e le varianze verso l'unità, favorendo rappresentazioni distribuite in modo calibrato nello spazio latente.

3.7 Il termine KL sulla variabile ausiliaria: $\mathcal{L}^{KL}_{z}$

Il quarto termine regolarizza il posteriore q_φ(z|sx) verso il prior standard $\mathcal{N}$(0, I_{dz}). Poiché q_φ(z|sx) = $\mathcal{N}$(z; μ^z_φ(sx), Σ^z_φ(sx)) con covarianza diagonale, la KL ha la stessa forma chiusa del termine precedente:

$$KL(q_\varphi(z|sx) \| \mathcal{N}(0, I_{dz})) = \tfrac{1}{2} \cdot \Sigma_{k=1}^{dz} [\Sigma^{(k)}\{z,\varphi\}(sx) + (\mu^{(k)}\{z,\varphi\}(sx))^2 - \log \Sigma^{(k)}_{z,\varphi}(sx) - 1]$$

Poiché questa quantità dipende da sx, che è a sua volta campionato da q_φ(sx|x), la perdita $\mathcal{L}^{KL}_{z}$ è:

$\mathcal{L}^{KL}_{z} = \mathbb{E}_{q_\varphi(sx|x)}\,[KL(q_\varphi(z|sx) \,\|\, \mathcal{N}(0, I_{dz}))] \approx KL(q_\varphi(z|\hat{sx}) \,\|\, \mathcal{N}(0, I_{dz}))$
con $\hat{sx} \sim q_\varphi(sx|x)$

stimata per Monte Carlo con un singolo campione $\hat{sx}$.

La semantica di $\mathcal{L}^{KL}_{z}$ *è diversa da quella di* $\mathcal{L}^{KL}_{sx}$: mentre la variabile sx ha un decodificatore esplicito (il termine $\mathcal{L}_{rec}$ ancora sx all'osservazione x), la variabile z non ha un decodificatore diretto nel modello generativo. Il solo vincolo che ne governa il comportamento è questo termine KL. La sua funzione è dunque quella di collo di bottiglia informativo: z deve comprimere la variabilità di sx rilevante per la previsione di sy, senza memorizzare idiosincrasie specifiche dei campioni di addestramento. Il prior standard impone che z sia distribuita in modo regolare, favorendo la generalizzazione del prior condizionale $p_\theta(sy|sx,z)$ a nuove osservazioni di contesto.

3.8 Il termine KL sulla rappresentazione bersaglio: $\mathcal{L}^{KL}_{sy}$

Il quinto termine è il più ricco di significato teorico, poiché è in esso che risiede la congiunzione formale tra l'inferenza variazionale e la previsione JEPA. La divergenza:

$\mathcal{L}^{KL}_{sy} = \mathbb{E}_{q_\varphi(sx,z|x)}\,[KL(q_\varphi(sy|sx,z,y) \,\|\, p_\theta(sy|sx,z))]$

misura la discrepanza tra il posteriore variazionale su sy — che ha accesso simultaneo a (sx, z) e all'osservazione bersaglio y — e il prior condizionale appreso $p_\theta(sy|sx,z)$ — che dipende unicamente da (sx,z) e non da y.

Per la parametrizzazione gaussiana diagonale, con:

$q_\varphi(sy|sx,z,y) = \mathcal{N}(sy;\ \mu^{sy}_\varphi(sx,z,y),\ \mathrm{diag}(\Sigma^{sy}_\varphi(sx,z,y)))$ $p_\theta(sy|sx,z) = \mathcal{N}(sy;\ \mu^{sy}_\theta(sx,z),\ \mathrm{diag}(\Sigma^{sy}_\theta(sx,z)))$

la KL tra due gaussiane diagonali è:

$$KL(q_\varphi \,\|\, p_\theta) = \tfrac{1}{2} \cdot \sum_{k=1}^{d} \left[\Sigma^{(k)}_q/\Sigma^{(k)}_p + (\mu^{(k)}_p - \mu^{(k)}_q)^2/\Sigma^{(k)}_p + \log(\Sigma^{(k)}_p/\Sigma^{(k)}_q) - 1\right]$$

dove si usa la notazione compatta $\Sigma^{(k)}_q = \Sigma^{(k)}_{sy,\varphi}(sx,z,y)$, $\Sigma^{(k)}_p = \Sigma^{(k)}_{sy,\theta}(sx,z)$, $\mu^{(k)}_q = \mu^{(k)}_{sy,\varphi}(sx,z,y)$ e $\mu^{(k)}_p = \mu^{(k)}_{sy,\theta}(sx,z)$.

La struttura di questo termine genera due pressioni simmetriche e complementari:

Rispetto ai parametri del codificatore bersaglio φ (che governano $\mu^{(k)}_q$ *e* $\Sigma^{(k)}_q$): *la minimizzazione di* $\mathcal{L}^{KL}_{sy}$ *tende ad avvicinare la media del posteriore alla media del prior condizionale —* $\mu^{(k)}_{sy,\varphi}(sx,z,y) \to \mu^{(k)}_{sy,\theta}(sx,z)$ *— e ad eguagliarne la varianza —* $\Sigma^{(k)}_{sy,\varphi} \to \Sigma^{(k)}_{sy,\theta}$. In altre parole, il codificatore bersaglio è spinto a produrre distribuzioni su sy coerenti con la previsione del predittore.

Rispetto ai parametri del prior condizionale θ (che governano μ^{(k)}*p e Σ^{(k)}p): la minimizzazione di ℒ^{KL}{sy} tende ad avvicinare la media del prior alla media del posteriore* — *μ^{(k)}*{sy,θ}(sx,z) → μ^{(k)}_{sy,φ}(sx,z,y) — e ad eguagliarne la varianza nella direzione favorevole alla riduzione della KL. In altre parole, il predittore è addestrato a produrre distribuzioni su sy che approssimano il più possibile il posteriore completo q_φ(sy|sx,z,y), che incorpora l'informazione sull'osservazione bersaglio.

Il bilanciamento tra queste due pressioni — mediato dalla tensione con il termine ℒ_gen che ancora q_φ(sy|sx,z,y) all'osservazione y — è il meccanismo centrale dell'apprendimento predittivo in Var-JEPA: il predittore p_θ(sy|sx,z) impara a «inseguire» il posteriore del codificatore bersaglio, che a sua volta è ancorato all'osservazione. Il risultato è un predittore che cattura la distribuzione condizionale di sy dato (sx,z), ovvero la struttura della relazione tra contesto e bersaglio nel senso probabilistico più generale.

3.9 Forma chiusa della KL e differenziazione automatica

L'implementazione pratica della perdita ℒ_ELBO sfrutta la calcolabilità in forma chiusa dei tre termini KL per ridurre la varianza dello stimatore a gradiente. In tutti e tre i casi — ℒ^{KL}_{sx}, ℒ^{KL}*z*, ℒ^{*KL*}{sy} — le distribuzioni coinvolte sono gaussiane (seppur con parametri che dipendono da reti neurali), e la formula chiusa evita la necessità di stimare le KL per campionamento.

Per il solo termine di stima di Monte Carlo — ℒ_rec e ℒ_gen — si utilizza il reparametrization trick di Kingma e Welling (2013). Per il campionamento sx ~ q_φ(sx|x) = $\mathcal{N}$(sx; μ^{sx}_φ(x), Σ^{sx}_φ(x)):

$$sx = \mu^{sx}_\varphi(x) + \Sigma^{sx}_\varphi(x)^{1/2} \cdot \varepsilon_{sx}, \quad \varepsilon_{sx} \sim \mathcal{N}(0, I_d)$$

Le operazioni ⊙ (prodotto componente per componente) sulle deviazioni standard σ^{sx}_φ(x) = diag(Σ^{sx}_φ(x))^{1/2} producono il campione sx in modo differenziabile rispetto a φ:

$$sx = \mu^{sx}_\varphi(x) + \sigma^{sx}_\varphi(x) \odot \varepsilon_{sx}$$

dove ε_sx è estratto casualmente ma trattato come costante ai fini della differenziazione. Lo stesso meccanismo si applica ai campioni di z e sy. Il gradiente rispetto ai parametri (θ, φ) è dunque calcolabile per retropropagazione automatica attraverso l'intero grafo computazionale, con un singolo campione per ciascuna variabile latente per minibatch.

3.10 Il ruolo dei coefficienti α: annealing e bilanciamento

I coefficienti α^rec, α^gen, α^{KL}_{sx}, α^{KL}*z e a^{KL}*{sy} nella perdita ℒ_ELBO non sono semplici iperparametri di peso: essi codificano scelte architetturali sulla relativa importanza dei cinque obiettivi parziali durante l'addestramento.

Una tecnica comune nei VAE per prevenire il collasso posteriore nelle fasi iniziali dell'ottimizzazione è il KL annealing: i termini KL vengono introdotti gradualmente, iniziando con pesi α^{KL} → 0 e aumentandoli linearmente fino ai valori target nel corso delle prime fasi di addestramento. Nell'implementazione Var-T-JEPA, Gögl e Yau adottano schedules di annealing lineari distinte per i tre termini KL:

α^{KL}{*sx*}*(t) = min(t / T^ann*{sx}, 1) · α^{KL}{*sx,final*} *a^{KL}z(t) = min(t / T^ann_z, 1) · a^{KL}{z,final}* *a^{KL}*{sy}(t) = min(t / T^ann_{sy}, 1) · α^{KL}_{sy,final}

dove t è il numero corrente di passi di ottimizzazione e T^ann_{·} è la durata dell'annealing per ciascun termine. Le durate ottimali dipendono dal dataset e dall'architettura, e vengono determinate empiricamente.

Quanto al bilanciamento dei pesi finali, Gögl e Yau riportano nella loro analisi di sensibilità (Tabella 4 della Appendice) che le prestazioni a valle restano robuste su un ampio intervallo di variazione dei coefficienti α, suggerendo che l'ELBO è relativamente insensibile alle scelte di ponderazione purché tutti i termini rimangano attivi. La variante di ablazione più critica rimane quella con α^rec = α^gen = 0 (variante G), dove la perdita di entrambi i termini di ricostruzione degrada severamente le rappresentazioni — confermando che la loro presenza è condizione necessaria, non opzionale.

3.11 Connessione con il β-VAE e interpretazione geometrica

Il modello Var-JEPA con coefficienti α arbitrari è concettualmente connesso al framework β-VAE di Higgins et al., dove si pesa la KL con un fattore β > 1 per incrementare la regolarizzazione e favorire la disentanglement delle rappresentazioni. In Var-JEPA, il ruolo analogo è svolto dai tre pesi α^{KL}{*sx*}*, a^{KL}z e a^{KL}{sy}, ciascuno dei quali controlla il grado di regolarizzazione del corrispondente spazio latente rispetto al prior. Un peso a^{KL}*{sx} elevato spinge sx verso il prior standard, comprimendo l'informazione mutua I(x; sx) e favorendo un collo di bottiglia più stretto — con potenziale beneficio per la generalizzazione ma rischio di perdita di informazione discriminativa.

La tensione tra i termini di ricostruzione (che vogliono sx informativo) e KL (che vogliono sx vicino al prior) è esattamente l'analogo della tensione rate-distortion della teoria dell'informazione. Il termine di ricostruzione corrisponde alla distortion D — la perdita di fedeltà nella codifica — mentre il termine KL corrisponde al rate R — la quantità di informazione necessaria per descrivere sx rispetto al prior. L'ELBO massimizzato corrisponde alla minimizzazione di una combinazione pesata D + β·R, il cui punto di equilibrio varia lungo la curva rate-distortion del dataset. Gögl e Yau non esplicitano questa connessione nel dettaglio, ma essa emerge naturalmente dalla struttura dell'ELBO e fornisce una prospettiva teorica aggiuntiva sulla scelta dei pesi α.

3.12 Riduzione della varianza nella stima del gradiente ELBO: tecniche avanzate

La stima Monte Carlo dell'ELBO attraverso il reparametrization trick produce stime non distorsate del gradiente rispetto ai parametri (θ, φ), ma può presentare varianza elevata nelle fasi iniziali dell'addestramento, quando le distribuzioni variazionali sono ancora distanti dall'ottimale. Comprendere le tecniche di riduzione della varianza disponibili per l'ELBO variazionale è essenziale per garantire la stabilità dell'addestramento di Var-JEPA su dataset di dimensioni realistiche.

La stima a singolo campione con reparametrization trick --- $s_x = \mu^{s_x}_\varphi(x) + \sigma^{s_x}_\varphi(x) \odot \varepsilon$, $\varepsilon \sim N(0, I)$ --- ha varianza $\mathrm{Var}[\nabla_\varphi \mathcal{L}_{ELBO}]$ proporzionale alla varianza della log-likelihood rispetto ai campioni ε. Per la componente di ricostruzione $\mathcal{L}_{rec}$, questa varianza è minimizzata quando $\sigma^{s_x}_\varphi(x) \to 0$ (regime deterministico), ma è massimizzata proprio nelle fasi iniziali quando il posterior è incerto. La stima Rao-Blackwellized del gradiente --- che sostituisce la media su ε con la media analitica ove possibile --- permette di ridurre la varianza per i termini KL (che sono calcolabili in forma chiusa) mantenendo la stima Monte Carlo solo per i termini di ricostruzione $\mathcal{L}_{rec}$ e $\mathcal{L}_{gen}$. Gögl e Yau adottano implicitamente questa tecnica usufruendo delle formule chiuse per le KL gaussiane (Sezione 3.9), ma non discutono esplicitamente la varianza dei gradienti per i termini di ricostruzione.

Un approccio alternativo è l'Importance Weighted Autoencoder (IWAE, Burda et al. 2015), che sostituisce la disuguaglianza di Jensen con un limite variazionale più stretto basato su K campioni invece di uno solo. L'obiettivo IWAE è $\mathcal{L}_{IWAE}^K = E_{s_1,...,s_K \sim q_\varphi(s|x,y)} [\log(1/K \Sigma_{k=1}^K p_\theta(x,y,s_k)/q_\varphi(s_k|x,y))]$, che converge verso la log-verosimiglianza marginale $\log p_\theta(x, y)$ per $K \to \infty$. La varianza del gradiente IWAE rispetto a φ decresce come $O(1/K)$, permettendo stime più accurate con soli K = 5-20 campioni aggiuntivi per batch. Il prezzo da pagare è un costo computazionale K volte superiore per le passate in avanti attraverso il decoder. Per Var-JEPA tabulare, dove i decoder sono MLP leggeri, questo overhead sarebbe accettabile; per versioni visive o linguistiche, richiederebbe un bilanciamento attento rispetto ai benefici.

Una terza tecnica, il leave-one-out (LOO) baseline dell'estimatore REINFORCE per le variabili discreti, non si applica direttamente al caso di Var-JEPA poiché tutte le variabili latenti sono continue e differenziabili attraverso il reparametrization trick. Tuttavia, nel caso di un'estensione di Var-T-JEPA a variabili tabulari categoriali (oltre alle variabili numeriche già gestite) attraverso tecniche di quantizzazione differenziabile o Gumbel-softmax, le tecniche LOO diventerebbero rilevanti per ridurre la varianza del gradiente rispetto ai parametri delle distribuzioni categoriali.

3.13 Bound più stretti e obiettivi variazionali alternativi per Var-JEPA

La scelta dell'ELBO come obiettivo variazionale in Var-JEPA, pur essendo la scelta canonica nella letteratura VAE, non è l'unica possibile. Diversi framework alternativi offrono bound più stretti sulla log-verosimiglianza o proprietà di stima con distorsione ridotta. Esaminare questi framework nell'ottica di Var-JEPA permette di identificare direzioni di ricerca per architetture variazionali più espressive.

Il legame diretto tra Var-JEPA e il framework ζ-VAE di Anokhin et al. (2021) è particolarmente illuminante. Il ζ-VAE introduce un parametro di temperatura ζ nella funzione obiettivo: $\mathcal{L}_{\zeta\text{-VAE}} = E_q [\log p(x|z)] - \zeta \cdot KL(q(z|x) || p(z))$. Per $\zeta < 1$, la penalizzazione KL è ridotta rispetto all'ELBO standard, favorendo posteriori più informativi a scapito della regolarità distributiva. Per $\zeta > 1$, viceversa, si ottiene una regolarizzazione più aggressiva analoga al β-VAE. La connessione con i coefficienti α di Var-JEPA è diretta: $\alpha^{KL}_{s_x}$ svolge il ruolo di ζ per il latente di contesto, α^{KL}_z per la variabile ausiliaria, e $\alpha^{KL}_{s_y}$ per il latente target. La differenza fondamentale è che in Var-JEPA i tre coefficienti possono essere regolati indipendentemente, permettendo un controllo più granulare della pressione regolarizzatrice su ciascun componente della rappresentazione.

Un framework ancora più espressivo è quello delle distribuzioni posteriori normalizing flow (NF-VAE), dove $q_\varphi(s|x)$ non è vincolato a essere gaussiano ma è parametrizzato attraverso una catena di trasformazioni biettive (flows) che mappano una distribuzione base semplice verso una distribuzione arbitrariamente complessa. Per Var-JEPA, questo sarebbe particolarmente rilevante per il posteriore target $q_\varphi(s_y \mid s_x, z, y)$: la distribuzione su s_y condizionata al bersaglio osservato y potrebbe essere multimodale (se più rappresentazioni di y sono compatibili con il contesto s_x), e una gaussiana diagonale potrebbe essere una rappresentazione inadeguata. L'uso di normalizing flows per il posteriore target permetterebbe di catturare questa multimodalità, al costo di un aumento della complessità computazionale e del numero di parametri. Gögl e Yau (2026) riconoscono implicitamente questa limitazione quando osservano che la distribuzione aggregata di s_y ha $KL_{agg} \approx 3.5$ rispetto a $N(0,I)$ nella variante A dell'ablation study --- un valore relativamente elevato che suggerisce una deviazione dall'isotropia gaussiana non catturabile dal prior condizionale gaussiano appreso.

Capitolo 4 — Prevenzione del collasso senza stop-gradient né EMA: analisi formale dei meccanismi variazionali

4.1 Il collasso come fenomeno di ottimizzazione: analisi della geometria del gradiente

Nel capitolo I si è descritto il collasso rappresentazionale come la degenerazione delle rappresentazioni verso soluzioni costanti che annullano la perdita JEPA senza estrarre informazione utile dai dati. Nel capitolo III si è mostrato, a livello qualitativo, perché l'ELBO di Var-JEPA sia strutturalmente incompatibile con tale degenerazione. Il presente capitolo approfondisce questa analisi attraverso lo studio formale della geometria del gradiente, la decomposizione ELBO surgery, e le evidenze sperimentali provenienti dallo studio di ablazione di Gögl e Yau (2026).

Il punto di partenza è la caratterizzazione precisa delle soluzioni collassanti nello spazio dei parametri (θ, φ). Una configurazione è «collassante» per il latente sx se esiste un vettore costante c_{sx} ∈ ℝ^d tale che μ^{sx}*φ(x)* = *c*{sx} per (quasi) ogni x nel supporto della distribuzione dei dati. Analogamente, una configurazione è collassante per sy se μ^{sy}*φ(sx,z,y)* ≈ *c*{sy} costante, e per z se μ^z_φ(sx) ≈ c_z costante. La forma più radicale di collasso è quella totale, in cui tutte e tre le medie sono costanti. Quella più insidiosa, come si vedrà, è il collasso parziale di sy, in cui il latente bersaglio degrada anche quando sx e z rimangono informativi.

4.2 Perché i termini di ricostruzione prevengono il collasso di sx

Si analizzi cosa accade a ℒ_ELBO quando il codificatore di contesto collassa: μ^{sx}*φ(x)* → *c*{sx} per ogni x, con Σ^{sx}*φ(x)* → *ε·I_d con ε → 0 (distribuzione sempre più concentrata sul punto c*{sx}).

Il termine ℒ^{KL}{*sx*} *si comporta come segue in questo limite. Con μ_q* = *c*{sx} e Σ_q = ε·I_d:

KL(𝒩(c_{sx}, ε·I_d) ‖ 𝒩(0, I_d)) = ½ · d·ε + ½ · ‖c_{sx}‖²_2 − ½·d·log(ε) − ½·d

Quando ε → 0, il termine −½·d·log(ε) → +∞: la KL diverge logaritmicamente al diminuire della varianza del posteriore. Questo rappresenta una penalità sempre crescente: un posteriore troppo concentrato (varianza → 0) è penalizzato dalla KL verso il prior gaussiano, che preferisce distribuzioni con varianza unitaria. Tuttavia, nel regime di c_{sx} = 0 e ε → 1, la KL tende a zero — quindi la KL da sola ammette la soluzione c_{sx} = 0, ε = 1 (posteriore coincidente con il prior standard) come minimo globale, indipendentemente dal contenuto di x.

È il termine ℒ_rec a rendere questa soluzione inadmissibile. Con sx campionato da q_φ(sx | x) ≈ 𝒩(c_{sx}, ε·I_d):

ℒ_rec ≈ (1/(2σ²_x)) · 𝔼_{ε_x ~ 𝒩(0,I)} [‖x − U^x_θ(c_{sx} + √ε·ε_x)‖²_2]

Quando ε → 0, sx ≈ c_{sx} con alta probabilità, e:

ℒ_rec → (1/(2σ²_x)) · ‖x − U^x_θ(c_{sx})‖²_2

Poiché U^x_θ(c_{sx}) è un vettore costante (indipendente da x), ‖x − U^x_θ(c_{sx})‖² è l'errore di approssimazione di una distribuzione non-triviale con una costante — che in generale è strettamente positivo per quasi ogni coppia (x, U^x_θ(c_{sx})). L'unico scenario

in cui $\mathcal{L}_{rec} \to 0$ con sx costante sarebbe quello in cui la distribuzione dei dati si concentra su un singolo punto $x = x_0$, nel qual caso si avrebbe $U^x_\theta(c_{sx}) = x_0$ e $\|x - U^x_\theta(c_{sx})\|^2 = 0$ per ogni x nella distribuzione. Ma questo scenario è escluso per qualsiasi dataset con variabilità non nulla.

L'argomento precedente può essere rafforzato analizzando la norma del gradiente di $\mathcal{L}$_ELBO rispetto ai parametri φ che governano $\mu^{sx}_\varphi(x)$. Il gradiente rispetto a $\mu^{sx}_\varphi(x)$ — inteso come funzione della variabile di ingresso x — ha la forma:

$\partial\mathcal{L}_ELBO/\partial\mu^{sx}\varphi(x) = a^{KL}_{sx} \cdot (\partial\mathcal{L}^{KL}_{sx}/\partial\mu^{sx}_\varphi) + \alpha^{rec} \cdot (\partial\mathcal{L}_rec/\partial\mu^{sx}\varphi) = a^{KL}_{sx} \cdot \mu^{sx}_\varphi(x) + \alpha^{rec} \cdot (-1/\sigma^2_x) \cdot (\partial U^x_\theta/\partial sx)^T \cdot (x - U^x_\theta(sx))$

Il termine KL spinge $\mu^{sx}\varphi(x)$ *verso 0 (poiché* $a^{KL}_{sx} \cdot \mu^{sx}_\varphi(x)$ punta nella direzione opposta a μ^{sx}_φ). Il termine di ricostruzione spinge $\mu^{sx}_\varphi(x)$ nella direzione del gradiente della ricostruzione, che dipende dall'ingresso x: direzioni diverse per ingressi diversi. Se $\mu^{sx}\varphi(x)$ *fosse costante in x, il gradiente di ricostruzione sarebbe lo stesso per ogni x, ma il decodificatore si adatterebbe a quel valore fisso e il residuo* ($x - U^x_\theta(c_{sx})$) sarebbe correlato con x attraverso la struttura della distribuzione dei dati — producendo gradienti sistematicamente non-nulli che spingono $\mu^{sx}_\varphi(x)$ in direzioni dipendenti da x. Il sistema non riesce a mantenere la costanza di $\mu^{sx}_\varphi(x)$ per le variazioni di x: il collasso è instabile rispetto al gradiente di $\mathcal{L}$_rec.

4.3 La decomposizione ELBO surgery e la regolarizzazione distributiva aggregata

Un'analisi più sottile della prevenzione del collasso distributivo — distinto dal collasso rappresentazionale — richiede la decomposizione ELBO surgery di Hoffman e Johnson (2016). Essa rivela la struttura interna del termine KL per campione e mostra come esso implichi automaticamente la regolarizzazione della distribuzione aggregata degli embedding.

La decomposizione si basa sull'identità seguente. Per qualsiasi coppia di variabili aleatorie (X, S) con distribuzione congiunta $p(X, S) = p(X) \cdot q_\varphi(S|X)$, e prior su S dato da p(S):

$\mathbb{E}_X [KL(q_\varphi(S|X) \| p(S))] = KL(q_\varphi(S) \| p(S)) + I_{q_\varphi}(X; S)$

dove $q_\varphi(S) = \int q_\varphi(S|X)\, p(X)\, dX$ è la distribuzione marginale (aggregata) di S sotto il modello variazionale, e $I_{q_\varphi}(X; S)$ è l'informazione mutua tra X e S sotto la distribuzione del modello:

$I_{q_\varphi}(X; S) = \mathbb{E}_{X,S \sim q_\varphi(S|X)\cdot p(X)} [\log(q_\varphi(S|X) / q_\varphi(S))] = \mathbb{E}_X [KL(q_\varphi(S|X) \| q_\varphi(S))] \geq 0$

Applicando questa decomposizione al termine KL su sx:

$\mathcal{L}^{KL}_{sx} = \mathbb{E}_x [KL(q_\varphi(sx|x) \| p(sx))] = KL(q_\varphi(sx) \| p(sx)) + I\{q_\varphi\}(x; sx) = KL(q_\varphi(sx) \| \mathcal{N}(0, I_d)) + I_{q_\varphi}(x; sx)$

con un'analoga decomposizione per il termine su z:

$\mathcal{L}^{KL}_{z} = \mathbb{E}_{x}[\mathbb{E}\{sx \sim q_\varphi(sx|x)\}[KL(q_\varphi(z|sx) \| p(z))]] = KL(q_\varphi(z) \| \mathcal{N}(0, I_{dz})) + I_{q_\varphi}(sx; z)$

dove $q_\varphi(z) = \iint q_\varphi(z|sx)\, q_\varphi(sx|x)\, p(x)\, dsx\, dx$ è la distribuzione aggregata di z.

Il significato di questa decomposizione è di duplice valenza. Il primo addendo in ciascuna espressione — $KL(q_\varphi(sx) \| \mathcal{N}(0, I_d))$ e $KL(q_\varphi(z) \| \mathcal{N}(0, I_{dz}))$ — misura la discrepanza tra la distribuzione marginale degli embedding e il prior standard: è precisamente il termine che SIGReg cerca di minimizzare direttamente attraverso il test di Epps-Pulley. La decomposizione ELBO surgery mostra dunque che minimizzare $\mathcal{L}^{KL}_{sx}$ implica automaticamente minimizzare $KL(q_\varphi(sx) \| \mathcal{N}(0, I_d))$, ovvero spingere la distribuzione aggregata degli embedding verso l'isotropia gaussiana. La regolarizzazione distributiva aggregata — che SIGReg impone esplicitamente come termine ausiliario — è una conseguenza implicita dei termini KL per campione dell'ELBO.

Il secondo addendo — $I_{q_\varphi}(x; sx)$ — è il collo di bottiglia informativo: esso misura quanta informazione relativa all'ingresso x sia contenuta nella rappresentazione latente sx. Valori elevati di $I_{q_\varphi}(x; sx)$ indicano che sx è molto informativo rispetto a x, il che è favorito dal termine di ricostruzione $\mathcal{L}_{rec}$. La minimizzazione della KL per campione comprime invece questa informazione mutua. Il punto di equilibrio tra i due termini — $\mathcal{L}_{rec}$ che vuole sx informativo e $\mathcal{L}^{KL}_{sx}$ che vuole sx compresso — determina il contenuto informativo ottimale della rappresentazione latente per il dataset dato.

4.4 L'asimmetria strutturale tra sx/z e sy: perché il collasso di sy è un caso diverso

Mentre la decomposizione ELBO surgery si applica in modo diretto ai termini KL su sx e z — producendo la regolarizzazione verso $\mathcal{N}(0, I)$ con prior fisso — il caso di sy è strutturalmente distinto. Il termine KL su sy è:

$\mathcal{L}^{KL}_{sy} = \mathbb{E}\{q_\varphi(sx,z|x)\}[KL(q_\varphi(sy|sx,z,y) \| p_\theta(sy|sx,z))]$

dove $p_\theta(sy|sx,z)$ non è un prior fisso ma il prior condizionale appreso. La decomposizione ELBO surgery nella forma analoga a quella per sx richiederebbe di considerare la distribuzione marginale aggregata di sy — $q_\varphi(sy) = \iiint q_\varphi(sy|sx,z,y)\, q_\varphi(z|sx)\, q_\varphi(sx|x)\, p(x,y)\, dx\, dy\, dsx\, dz$ — e la divergenza verso la distribuzione marginale del prior condizionale — $p_\theta(sy) = \iint p_\theta(sy|sx,z)\, q_\varphi(sx,z|x)\, p(x)\, dx\, dsx\, dz$. Ma $p_\theta(sy)$ non è $\mathcal{N}(0, I_d)$: è la distribuzione marginale predittiva, che dipende dai parametri θ e non è in generale gaussiana.

La conseguenza è che la distribuzione aggregata di sy non è naturalmente spinta verso $\mathcal{N}(0, I_d)$ dai soli termini KL dell'ELBO. Questo spiega la differenza empirica osservata nella Tabella 1 di Gögl e Yau: $KL_{agg}(q(sx) \| \mathcal{N}(0, I)) = 0.113 \pm 0.037$ (vicino a zero), mentre $KL_{agg}(q(sy) \| \mathcal{N}(0, I)) = 3.530 \pm 0.377$ (significativamente distante). Questa asimmetria non è un difetto del modello: è una conseguenza teoricamente attesa della struttura diversa dei due termini KL.

Tentare di correggere questa asimmetria aggiungendo SIGReg esplicita su sy (variante C nella Tabella 1: λ_{sx} = 0, λ_{sy} = 10) riduce KL_agg(q(sy)) da 3.530 a 1.999, avvicinando la distribuzione aggregata di sy all'isotropia gaussiana. Tuttavia, l'accuratezza della sonda lineare rimane 0.993 ± 0.002 — identica alla variante A senza SIGReg su sy — confermando che tale regolarizzazione aggiuntiva non porta benefici discriminativi e altera la geometria naturale di sy senza miglioramento pratico. La distribuzione aggregata di sy è correttamente governata dal prior condizionale appreso, non dal prior standard.

4.5 Il ruolo indispensabile di $\mathcal{L}^{KL}_{sx}$: evidenza sperimentale

La variante di ablazione (E) nella Tabella 1 di Gögl e Yau — in cui $\alpha^{KL}_{sx} = 0$, con tutti gli altri termini mantenuti ai valori predefiniti — fornisce la prova più diretta dell'importanza della regolarizzazione KL sul latente di contesto. I diagnostici distributivi in questa variante mostrano un deterioramento drammatico:

KL_agg(q(sx) ‖ $\mathcal{N}$(0,I)) = 8.374 ± 1.944 (vs. 0.113 ± 0.037 nella variante A) SIGReg-MSE(sx) = 5.1×10^{-2} ± 1.9×10^{-2} (vs. 4.0×10^{-4} ± 1.0×10^{-4} nella variante A) ‖Cov(sx) − I‖_F = 7.197 ± 1.522 (vs. 0.649 ± 0.127 nella variante A) $\|\mathbb{E}[sx]\|_2$ = 2.550 ± 0.607 (vs. 0.082 ± 0.009 nella variante A)

Nonostante questo collasso distributivo severo — con la distribuzione aggregata di sx distante dall'isotropia gaussiana, con media lontana dall'origine e covarianza fortemente non-identità — l'accuratezza della sonda lineare rimane 0.996 ± 0.002, identica alla variante completa. Questo risultato apparentemente paradossale rivela una tensione fondamentale: le rappresentazioni sono discriminativamente accurate anche in assenza di regolarizzazione distributiva, ma la struttura geometrica dello spazio latente risulta severamente non-gaussiana.

La rilevanza della regolarizzazione KL su sx dipende dunque dall'applicazione a valle: per compiti di classificazione con sonda lineare su rappresentazioni arbitrarie, la distribuzione aggregata degli embedding è irrilevante — ciò che conta è la separabilità lineare delle classi, che si preserva anche con embedding distribuiti in modo non-gaussiano. Per compiti che richiedono una struttura metrica ben calibrata nello spazio latente — ricerca per vicinanza, generazione per interpolazione, transfer learning — la non-gaussianità della distribuzione aggregata potrebbe compromettere le prestazioni in modo significativo. Per la valutazione selettiva basata sull'incertezza (risk-coverage curves), la distribuzione dei valori di varianza posteriore dipende dalla regolarizzazione: in assenza di $\mathcal{L}^{KL}_{sx}$, le varianze posteriori potrebbero assumere valori arbitrari, degradando la calibrazione dell'incertezza.

4.6 Prevenzione del collasso senza stop-gradient né EMA: il meccanismo complessivo

Raccogliendo i fili dell'analisi, si può ora descrivere il meccanismo complessivo attraverso cui Var-JEPA previene il collasso rappresentazionale senza ricorrere né allo stop-gradient né all'EMA. Il meccanismo è plurilivello e si articola su tre strati distinti.

Il primo strato è la prevenzione del collasso informativo di sx, realizzata dalla tensione tra $\mathcal{L}$_rec e $\mathcal{L}^{KL}\{sx\}$. *Il termine di ricostruzione $\mathcal{L}$_rec impedisce a sx di diventare costante in x (poiché una rappresentazione costante produce una ricostruzione pessima per ogni x non-triviale), mentre il termine KL* $\mathcal{L}^{KL}\{sx\}$ impedisce a sx di divergere verso rappresentazioni arbitrariamente complesse (poiché distribuzioni molto lontane dal prior standard vengono penalizzate). Il punto di equilibrio è una rappresentazione sx che cattura l'informazione di x in modo calibrato.

Il secondo strato è la prevenzione del collasso informativo di sy, realizzata dalla tensione tra $\mathcal{L}$_gen e $\mathcal{L}^{KL}\{sy\}$. *Il termine generativo $\mathcal{L}$_gen impedisce a sy di diventare indipendente da y (poiché una rappresentazione costante produce una ricostruzione pessima del bersaglio), mentre il termine KL* $\mathcal{L}^{KL}\{sy\}$ spinge il posteriore su sy verso il prior condizionale appreso $p_\theta(sy|sx,z)$ — il predittore. Il punto di equilibrio è un posteriore su sy che incorpora sia l'informazione di y (pressione di $\mathcal{L}$_gen) sia la struttura predittiva del contesto (pressione di $\mathcal{L}^{KL}_{sy}$).

Il terzo strato è la prevenzione del collasso del predittore, realizzata dall'interazione tra $\mathcal{L}^{KL}_{sy}$ e la struttura generativa. Il predittore $p_\theta(sy|sx,z)$ — cui corrisponde la componente generativa $\mu^{sy}_\theta(sx,z)$, $\Sigma^{sy}\theta(sx,z)$ — *è addestrato a minimizzare* $\mathcal{L}^{KL}\{sy\}$ rispetto a θ. Questo richiede che il prior condizionale approssimi il più possibile il posteriore $q_\varphi(sy|sx,z,y)$. Se il predittore collassasse verso una distribuzione costante — $\mu^{sy}_\theta(sx,z) = c$ per ogni (sx,z) — la KL con il posteriore sarebbe elevata per ogni coppia (sx,z) tale che $\mu^{sy}_\varphi(sx,z,y) \neq c$, e il gradiente rispetto a θ spingerebbe il predittore ad adattarsi alla struttura delle coppie (sx,z,y). Dunque, anche il collasso del predittore è penalizzato strutturalmente dall'ELBO.

4.7 Confronto formale con stop-gradient ed EMA: cosa guadagna il framework variazionale

Si può ora fornire una caratterizzazione formale precisa di ciò che distingue il framework variazionale dai meccanismi deterministici di anti-collasso. Tre aspetti meritano di essere messi in evidenza.

Dal punto di vista dell'obiettivo di ottimizzazione: il sistema Var-JEPA minimizza un'unica funzione obiettivo $\mathcal{L}$_ELBO attraverso discesa del gradiente su tutti i parametri (θ, φ) simultaneamente. Non esistono blocchi di gradiente, aggiornamenti asimmetrici, o dinamiche ibride. Il sistema ha garanzie di convergenza standard per metodi di ottimizzazione del primo ordine su obiettivi stocastici regolari — garanzie che la combinazione stop-gradient/EMA non possiede per l'assenza di un obiettivo unificato.

Dal punto di vista dello spazio delle soluzioni: la perdita $\mathcal{L}$_ELBO non ha minimi globali collassanti compatibili con dati non-triviali. Come dimostrato nelle sezioni precedenti, qualsiasi configurazione collassante produce valori elevati di almeno uno tra $\mathcal{L}$_rec e $\mathcal{L}$_gen. Nella JEPA deterministica, invece, le soluzioni collassanti sono minimi globali legittimi della perdita $\mathcal{L}$_JEPA: stop-gradient ed EMA le rendono localmente instabili o difficilmente raggiungibili, ma non le eliminano dallo spazio delle soluzioni. Var-JEPA ridefinisce lo spazio delle soluzioni in modo che le configurazioni collassanti non siano minimizzanti.

Dal punto di vista degli iperparametri: stop-gradient non introduce iperparametri ma richiede la scelta della direzione del blocco (quale ramo bloccare), che deve essere decisa caso per caso. L'EMA introduce il coefficiente τ, la cui scelta influenza in modo critico la stabilità e la velocità di convergenza del sistema. In Var-JEPA, i coefficienti α e i tempi di annealing T^ann sono iperparametri aggiuntivi, ma la loro influenza è più principiata — essi modulano il peso relativo di termini con significato probabilistico trasparente — e l'analisi di sensibilità di Gögl e Yau mostra che il sistema è robusto rispetto a variazioni moderate di questi parametri intorno ai valori ottimali.

4.8 Il collasso distributivo e SIGReg come regolarizzazione complementare

Resta da analizzare il caso in cui la prevenzione del collasso distributivo di sy — quella legata alla distribuzione aggregata di sy rispetto a $\mathcal{N}(0, I_d)$ — venga perseguita attraverso l'aggiunta di SIGReg sul latente bersaglio. Gögl e Yau esaminano la variante (C) (λ_{sx} = 0, λ_{sy} = 10) e trovano che essa riduce KL_agg(q(sy)) da 3.530 a 1.999, avvicinando la distribuzione aggregata di sy all'isotropia gaussiana. Tuttavia, come già notato, l'accuratezza non migliora.

Questo risultato ha un'interpretazione teorica precisa. La distribuzione aggregata ottimale di sy per il downstream probing è quella che massimizza le prestazioni delle sonde a valle — classificatori lineari, regressori, etc. Balestriero e LeCun (2025) dimostrano in LeJEPA che la distribuzione isotropa gaussiana è minimax-ottimale sotto una classe ampia di sonde, ma questa ottimalità vale per sx — la rappresentazione di contesto usata come feature per i task a valle — non necessariamente per sy, che nel protocollo di Var-T-JEPA viene effettivamente usata come feature. Poiché Gögl e Yau usano le medie posteriori sw = μ^{sy}*φ(sx,z,w) come embedding finali per l'addestramento dei predittori a valle, la struttura di queste medie è ciò che conta per le prestazioni downstream — e questa struttura è già ottimizzata dal termine $\mathcal{L}$^{KL}{sy} verso il prior condizionale appreso, non verso il prior standard. La regolarizzazione SIGReg su sy spinge le medie posteriori verso una distribuzione gaussiana isotropa, ma se il prior condizionale appreso non ha questa forma, tale forzatura può essere subottimale.

4.9 Dimostrazione quantitativa: la catena del collasso bloccata

Per rendere concrete le argomentazioni teoriche precedenti, si ripercorre la sequenza delle varianti di ablazione nella Tabella 1 di Gögl e Yau (2026) come sequenza di «esperimenti mentali» che smontano progressivamente i meccanismi anti-collasso.

La variante (A) — ELBO completo, tutti i termini attivi — rappresenta il sistema integro: accuratezza 0.996 ± 0.002 su sx, 0.993 ± 0.001 su sy. Le proprietà distributive sono eccellenti per sx (KL_agg = 0.113) e accettabili per sy data la struttura del prior condizionale (KL_agg = 3.530).

La variante (E) — $\mathcal{L}$^{KL}_{sx} rimosso — mantiene l'accuratezza (0.996) ma degrada la distribuzione aggregata di sx (KL_agg = 8.374). Il meccanismo di prevenzione del collasso

distributivo di sx è disattivato: sx rimane informativo (grazie a $\mathcal{L}$_rec) ma la sua distribuzione aggregata non è isotropa.

La variante (F) — $\mathcal{L}^{KL}_{sy}$ rimosso — mantiene l'accuratezza su sx (0.996) ma degrada sia l'accuratezza su sy (0.983) sia la distribuzione di sy (KL_agg = 6.201). Con il termine KL su sy rimosso, il posteriore $q_\varphi(sy|sx,z,y)$ non è più spinto verso il prior condizionale, e il predittore non impara più a anticipare sy. Il sistema degenera verso un semplice encoder del bersaglio senza accoppiamento predittivo con il contesto.

La variante (G) — $\mathcal{L}$_rec e $\mathcal{L}$_gen entrambi rimossi — produce la degenerazione più drammatica: accuratezza 0.571 ± 0.010 su sx e 0.543 ± 0.033 su sy — prossima alla casualità per un problema a tre classi. I termini KL sono ancora attivi e producono distribuzioni molto vicine al prior standard (KL_agg(sx) = 0.015, KL_agg(sy) = 0.055), ma le rappresentazioni sono semanticamente vuote: avere la distribuzione «giusta» non è sufficiente se le rappresentazioni non portano informazione sull'ingresso.

La variante (I) — tutti i termini KL rimossi, solo $\mathcal{L}$_rec e $\mathcal{L}$_gen attivi — produce accuratezza 0.995 ± 0.002 su sx (ottima) ma distribuzione fortemente non-gaussiana: KL_agg(sx) = 10.465 ± 2.207. Le rappresentazioni sono informative ma non regolate: il sistema apprende embeddings discriminativi ma distribuitisi arbitrariamente nello spazio latente.

Infine, la variante (J) — tutti i termini KL rimossi ma SIGReg aggiunta — mostra accuratezza 0.996 ± 0.002 (recuperando la qualità della variante A) e distribuzione migliorata rispetto a (I) ma ancora non-gaussiana (KL_agg(sx) = 2.115). Ciò conferma che SIGReg può parzialmente compensare l'assenza dei termini KL per la struttura distributiva aggregata, ma non per la qualità discriminativa delle singole rappresentazioni — che dipende interamente dai termini di ricostruzione.

La sequenza di queste varianti delinea con chiarezza la struttura del framework anti-collasso in Var-JEPA: i termini di ricostruzione ($\mathcal{L}$_rec, $\mathcal{L}$_gen) sono responsabili dell'informatività delle rappresentazioni; i termini KL sono responsabili della struttura distributiva dello spazio latente; e la loro combinazione nell'ELBO completo produce il sistema ottimale — informativo e geometricamente regolato — senza la necessità di alcun meccanismo euristico esterno.

4.10 Analisi della convergenza: garanzie teoriche per l'ottimizzazione dell'ELBO variazionale

L'analisi formale della convergenza dell'ottimizzazione dell'ELBO in Var-JEPA richiede di considerare due aspetti distinti: la convergenza dell'ottimizzazione stocastica rispetto ai parametri (θ, φ), e la qualità del punto fisso raggiunto in termini di approssimazione del vero posteriore. Questi due aspetti sono concettualmente separati, ma interagiscono in modo non banale nella pratica.

Per la convergenza dell'ottimizzazione stocastica, i risultati standard per Adam (Kingma e Ba, 2014) e varianti garantiscono la convergenza verso un punto stazionario (gradiente nullo) per funzioni non convesse con gradiente Lipschitz, a condizione che il learning

rate segua una schedule decrescente appropriata o che venga usato un learning rate costante sufficientemente piccolo. Per $\mathcal{L}$_ELBO, la condizione di Lipschitz del gradiente è soddisfatta localmente per reti neurali con pesi limitati, e la condizione di variance bound (condizione (C2) nella letteratura SGD) è soddisfatta per i termini KL in forma chiusa e, con varianza finita, per i termini di ricostruzione stimati via Monte Carlo. I risultati empirici di Gögl e Yau mostrano una convergenza stabile in 200 epoche su tutti i dataset tabulari testati, coerente con le garanzie teoriche per Adam con KL annealing.

Più sottile è la questione della qualità del punto fisso. L'ELBO è un lower bound stretto (tight lower bound) sulla log-verosimiglianza solo se $q_\varphi(s_x, z, s_y \mid x, y)$ coincide con il vero posteriore $p_\theta(s_x, z, s_y \mid x, y)$. Per reti neurali di capacità finita, questo non accade mai esattamente, e il gap variazionale $KL(q_\varphi \,||\, p_\theta(\cdot|x,y)) > 0$ rimane positivo. La questione è quanto grande sia questo gap in pratica. Un risultato di Turner e Sahani (2011) stabilisce che per modelli con variabili latenti continue e famiglie variazionali gaussiane, il gap è proporzionale alla "curvatura" del vero posteriore: posteriori multimodali o a coda pesante producono gap più grandi di posteriori unimodali quasi-gaussiani. Per i dati tabulari analizzati da Gögl e Yau, la struttura del posteriore è presumibilmente sufficientemente regolare da mantenere il gap piccolo, ma per domini più complessi (immagini ad alta risoluzione, sequenze linguistiche lunghe) questo potrebbe diventare un fattore limitante.

Un'ulteriore considerazione riguarda il fenomeno del posterior collapse nel regime di alta capacità. Quando il decoder è molto espressivo, la log-verosimiglianza $\mathcal{L}$_rec può essere massimizzata indipendentemente dal contenuto di s_x, rendendo la pressione verso la ricostruzione insufficiente a "attivare" il canale latente. Lucas et al. (2019) dimostrano che questo fenomeno è aggravato dalla presenza del termine KL che spinge s_x verso il prior, e propongono tecniche di masked KL annealing in cui la KL dei singoli neuroni latenti viene introdotta sequenzialmente. Per Var-JEPA, il posterior collapse del latente z --- che non ha un decodificatore diretto ma solo la pressione indiretta attraverso il prior condizionale appreso --- è particolarmente a rischio in questa situazione. I risultati degli esperimenti di ablazione di Gögl e Yau mostrano che la variante (F) (senza KL su s_y) porta a una degradazione dell'accuratezza da 0.993 a 0.983, suggerendo che il canale s_y rimane attivo anche senza la pressione KL diretta; ma non forniscono dati analoghi per il canale z, che rimane una questione empiricamente poco esplorata.

4.11 Confronto formale con i metodi di SSL contrastivo: SimCLR, MoCo e BYOL come casi speciali

I metodi di self-supervised learning contrastivi --- SimCLR (Chen et al. 2020), MoCo (He et al. 2020), BYOL (Grill et al. 2020) --- e la famiglia JEPA condividono l'obiettivo di apprendere rappresentazioni invarianti rispetto a trasformazioni semanticamente irrilevanti, ma differiscono radicalmente nella struttura della funzione obiettivo e nelle proprietà delle rappresentazioni apprese. Un confronto formale tra questi approcci è illuminante sia per comprendere le specificità di Var-JEPA sia per identificare le condizioni sotto cui ciascun paradigma è preferibile.

SimCLR minimizza la InfoNCE loss su coppie di viste aumentate: $\mathcal{L}_{SimCLR} = -\log(\exp(\text{sim}(z_i, z_j)/\tau) / \Sigma_{k\neq i} \exp(\text{sim}(z_i, z_k)/\tau))$, dove z_i, z_j sono le rappresentazioni di due viste della stessa immagine e k itera su tutti i campioni del batch. Questa perdita può essere vista come l'apprendimento di una distribuzione di similarità sul batch: la rappresentazione di ciascun campione deve essere più simile (in termini di coseno normalizzato) alla propria vista alternativa che a quelle di altri campioni. Il problema fondamentale di questo approccio è la dipendenza dalla dimensione del batch: per batch piccoli, il numero di negativi è limitato e la stima del denominatore è noisosa. JEPA evita questo problema usando la predizione nello spazio latente anziché la contrastività esplicita.

BYOL, come notato nelle sezioni precedenti, è il precursore diretto di JEPA in termini di meccanismo anti-collasso attraverso EMA. La connessione formale tra BYOL e Var-JEPA può essere precisata come segue: BYOL implementa implicitamente un ELBO degradato in cui (a) il termine di ricostruzione del contesto $\mathcal{L}_{rec}$ è assente (non c'è decoder per il contesto), (b) il termine KL su s_x è sostituito dall'EMA del target encoder come regolarizzatore distributivo implicito, e (c) il termine KL su s_y è la perdita MSE tra la previsione e la rappresentazione target. Gögl e Yau dimostrano che questo "ELBO degradato" è instabile senza il termine di generazione $\mathcal{L}_{gen}$, la cui assenza è compensata dallo stop-gradient --- un compromesso strutturalmente imperfetto che produce la fragilità documentata nel Capitolo 1. La gerarchia formale è dunque: Var-JEPA (ELBO completo) $\supset$ JEPA canonico (ELBO degradato + stop-gradient + EMA) $\supset$ BYOL (caso speciale di JEPA canonico con encoder bersaglio aggiornato solo via EMA).

Capitolo 5 — Var-T-JEPA per dati tabulari: generalizzazione del paradigma

5.1 La specificità del dominio tabulare

Trasferire un paradigma architetturale nato per la visione artificiale — ove la struttura locale dei dati è sfruttata attraverso maschere spaziali o temporali di patch contigui — verso il dominio dei dati tabulari implica una revisione non meramente ingegneristica, bensì concettuale. I dati tabulari si distinguono da immagini e sequenze audio-visive per almeno tre caratteristiche fondamentali che condizionano in modo radicale le scelte di progettazione dell'architettura e della funzione obiettivo.

In primo luogo, la natura intrinsecamente eterogenea degli attributi: ogni colonna di una tabella può veicolare un tipo semantico distinto — variabili numeriche continue, variabili categoriali nominali, variabili ordinali — ciascuna con scala, distribuzione marginale e cardinalità proprie. Non esiste, pertanto, una rappresentazione omogenea a livello di pixel che consenta di trattare uniformemente tutti i costituenti della tupla. Questa eterogeneità impone di adottare strategie di tokenizzazione specifiche per tipologia, sì da proiettare ogni attributo

in uno spazio latente condiviso prima di qualunque elaborazione congiunta attraverso meccanismi di attenzione.

In secondo luogo, l'assenza di una struttura di vicinanza spaziale o temporale. Nelle immagini, patch adiacenti condividono informazione locale; nei video, frame consecutivi presentano correlazioni temporali sfruttabili dalla predizione in spazio latente. Nelle tabelle, la posizione di una colonna è convenzionale e priva di significato geometrico: l'attributo "reddito" non è "vicino" all'attributo "età" in alcun senso diverso dalla disposizione arbitraria delle colonne nel foglio originale. Ciò implica che le strategie di mascheratura non possono appoggiarsi alla contiguità: occorre definire una partizione casuale delle colonne in sottoinsieme contestuale e sottoinsieme bersaglio, rinunciando all'intuizione spaziale che guida l'I-JEPA o il V-JEPA.

In terzo luogo, le interazioni tra attributi sono spesso non ovvie, non lineari e difficilmente catturabili da modelli parametrici a struttura fissa. Proprio questa ricchezza di dipendenze latenti rende il compito di predizione in spazio di rappresentazione particolarmente fertile per l'apprendimento non supervisionato: un modello costretto a ricostruire la rappresentazione di un sottoinsieme di attributi a partire da quella di un sottoinsieme complementare deve necessariamente apprendere le covarianze strutturali tra le variabili, acquisendo così una comprensione della geometria dello spazio dei dati che va oltre la semplice memorizzazione dei valori marginali.

Var-T-JEPA nasce dall'esigenza di istanziare il framework variazionale descritto nei capitoli precedenti in questo dominio specifico, adottando come punto di riferimento l'architettura deterministica T-JEPA di Thimonier et al. (2025), che aveva già adattato l'obiettivo JEPA alla predizione in spazio di feature per dati eterogenei, ma senza incorporare la struttura probabilistica latente che il nuovo paradigma variazionale richiede.

5.2 La strategia di mascheratura a livello di attributo

Il meccanismo di mascheratura in Var-T-JEPA opera direttamente sulla dimensione degli attributi piuttosto che su una dimensione spaziale. Data una tupla $w \in \mathbb{R}^{D_{feat}}$, si costruisce una maschera contestuale $m_{ctx} \in \{0,1\}^{D_{feat}}$ e K maschere bersaglio $m_{trg}^{(k)} \in \{0,1\}^{D_{feat}}$, con $k = 1, \ldots, K$. La vista contestuale e le viste bersaglio sono definite rispettivamente come:

$$x = m_{ctx} \odot w, \quad \|m_{ctx}\|_0 = M_{ctx}$$

$$y_k = m_{trg}^{(k)} \odot w, \quad \|m_{trg}^{(k)}\|_0 = M_{trg}, \quad k = 1, \ldots, K$$

dove $\odot$ denota il prodotto elemento per elemento e $\|\cdot\|_0$ conta il numero di elementi non nulli, equivalente qui alla cardinalità delle colonne selezionate. Le dimensioni delle maschere non sono fisse ma campionate uniformemente a ogni passo di ottimizzazione:

$$M_{ctx} \sim \text{Uniform}[\lfloor D_{feat} \cdot r_{ctx}^{min} \rfloor, \lfloor D_{feat} \cdot r_{ctx}^{max} \rfloor]$$

$$M_{trg} \sim \text{Uniform}[\lfloor D_{feat} \cdot r_{trg}^{min} \rfloor, \lfloor D_{feat} \cdot r_{trg}^{max} \rfloor]$$

con il vincolo $M_{ctx} + M_{trg} \leq D_{feat}$, che garantisce la non sovrapposizione tra contesto e bersaglio durante l'addestramento. Questa variabilità casuale nelle dimensioni delle maschere ha un effetto regolarizzante: il modello non può adattarsi a una particolare partizione fissa e deve sviluppare rappresentazioni flessibili, capaci di integrare informazione da sottoinsiemi di attributi di cardinalità variabile.

La scelta di generare K predizioni bersaglio per ogni campione introduce ridondanza costruttiva: il modello deve produrre K distribuzioni latenti distinte, ciascuna condizionata al medesimo contesto ma corrispondente a un diverso sottoinsieme mascherato. Questa molteplicità di obiettivi condizionali arricchisce il segnale di addestramento e favorisce la generalizzazione.

5.3 La tokenizzazione degli attributi eterogenei

Prima che qualunque meccanismo di codifica possa operare, ogni attributo della tupla deve essere proiettato in uno spazio vettoriale comune di dimensione D_{hid}. La procedura di tokenizzazione differisce necessariamente per gli attributi numerici e per quelli categoriali.

Per le variabili numeriche $j \in I_{num}$, si utilizza una proiezione lineare condivisa $W_{num} \in \mathbb{R}^{D_{hid} \times 1}$ combinata con un termine di bias specifico per attributo $b^{(j)} \in \mathbb{R}^{D_{hid}}$, che consente di modellare le differenze di scala e di distribuzione marginale tra colonne diverse. Il token risultante incorpora inoltre un embedding di tipo $e_{type}^{num} \in \mathbb{R}^{D_{hid}}$, che segnala al codificatore l'appartenenza alla categoria numerica, e un embedding posizionale $e_{pos}^{(j)} \in \mathbb{R}^{D_{hid}}$, che codifica la posizione della colonna nella sequenza:

$$t_{ctx}^{(j)} = W_{num} \cdot x^{(j)} + b^{(j)} + e_{type}^{num} + e_{pos}^{(j)}, j \in I_{num}$$

Per le variabili categoriali $j \in I_{cat}$ con L_j categorie distinte, si ricorre a una tabella di embedding dedicata $Embed_{L_j} : \{0, 1, \ldots, L_j - 1\} \rightarrow \mathbb{R}^{D_{hid}}$, combinata con i medesimi embedding di tipo e_{type}^{cat} e posizionali:

$$t_{ctx}^{(j)} = Embed_{L_j}(x^{(j)}) + e_{type}^{cat} + e_{pos}^{(j)}, j \in I_{cat}$$

La presenza di embedding di tipo distinti per variabili numeriche e categoriali è fondamentale: anche in assenza di qualunque altra informazione, il modello può così inferire la modalità di elaborazione appropriata per ciascun token, sfruttando la distinzione semantica già al livello della rappresentazione iniziale.

5.4 I sei componenti della rete variazionale

L'architettura di Var-T-JEPA si articola in sei moduli funzionalmente distinti, ciascuno dei quali implementa un aspetto specifico del framework variazionale: due codificatori (contestuale e ausiliario), il posteriore bersaglio, il predittore e due decodificatori (contestuale e bersaglio). La separazione modulare rispecchia la fattorizzazione del posteriore variazionale derivata nel capitolo 3 e assicura che i vincoli di non fuga dell'informazione siano rispettati architetturalmente, non solo attraverso la funzione obiettivo.

Il codificatore contestuale. Il modulo $q_\varphi(s_x^{(j)}|x)$ elabora la sequenza di token mascherati attraverso un transformer standard con meccanismi di auto-attenzione multi-testa e strati feed-forward. Seguendo la convenzione di T-JEPA, vengono preposti alla sequenza N_{CLS} token di classificazione apprendibili, che fungono da aggregatori globali dell'informazione contestuale. L'uscita del transformer è una sequenza contestualizzata:

$$h_{ctx} = \text{Transformer}([\{t_{CLS}^{(i)}\}_{i=1}^{N_{CLS}}, \{t_{ctx}^{(j)}\}_{j \in m_{ctx}}]) \in \mathbb{R}^{(N_{CLS} + M_{ctx}) \times D_{hid}}$$

Ciascun token di questa sequenza viene quindi proiettato linearmente per produrre i parametri della distribuzione posteriore variazionale:

$$\mu_{sx_\varphi}(h_{ctx}) = \text{Linear}(h_{ctx})$$

$$\log \Sigma_{sx_\varphi}(h_{ctx}) = \text{Linear}(h_{ctx})$$

$$q_\varphi(s_x^{(j)}|x) = \mathcal{N}(s_x^{(j)}; \mu_{sx_\varphi}(h_{ctx})^{(j)}, \Sigma_{sx_\varphi}(h_{ctx})^{(j)})$$

La scelta di parametrizzare la varianza attraverso il logaritmo è una tecnica standard che garantisce la positività della varianza indipendentemente dai valori prodotti dalla rete, senza necessità di attivazioni vincolate.

Il codificatore ausiliario. Il modulo $q_\varphi(z|s_x)$ inferisce i latenti ausiliari dalla rappresentazione contestuale aggregata $\bar{s}_x$, ottenuta attraverso un meccanismo di pooling attenzionale invariante all'ordine. Il pooling attenzionale è preferito al semplice pooling medio perché consente al modello di selezionare dinamicamente quali token contestuali sono più informativi per la sintesi della variabile ausiliaria z. L'elaborazione avviene attraverso un piccolo percettrone multistrato:

$$h_{aux}^{(l)} = \text{GELU}(\text{LayerNorm}(\text{Linear}(h_{aux}^{(l-1)})))$$

$$\mu_{z_\varphi}, \log \sigma_{z_\varphi} = \text{MLP}_\varphi(\bar{s}_x)$$

$$q_\varphi(z|s_x) = \mathcal{N}(z; \mu_{z_\varphi}, \text{diag}(\sigma_{z_\varphi})^2)$$

La rete ausiliaria adotta deliberatamente una capacità inferiore rispetto al codificatore principale: z non deve diventare una rappresentazione elaborata del contesto, bensì un segnale predittivo leggero che cattura la variabilità residua di s_y non spiegata da s_x. Questo design è coerente con il ruolo assegnato a z nella generazione: z è campionato dalla gaussiana standard $p(z) = \mathcal{N}(0, I)$ al momento dell'inferenza generativa, e deve quindi corrispondere a una sorgente di rumore interpretabile piuttosto che a una compressa dell'intera informazione contestuale.

Il posteriore bersaglio. Il modulo $q_\varphi(s_y^{(j)}|s_x, z, w; m_{trg}^{(k)})$ è il componente di maggiore complessità, poiché deve integrare tre fonti di informazione eterogenee: i latenti contestuali s_x, la variabile ausiliaria z e i valori grezzi degli attributi w. La sua realizzazione segue un'architettura transformer analoga al codificatore contestuale, ma estesa per incorporare i latenti come token speciali preposti alla sequenza.

I latenti aggregati s̄_x e z vengono proiettati nello spazio degli embedding e dotati di embedding di tipo distinti che ne segnalano il ruolo:

t_{s̄_x}^{(i)} = W_{s_x} · s̄_x^{(i)} + e_type^{pooled-ctx}, i = 1, …, L_pool

t_z = W_z · z + e_type^{aux-latent}

La sequenza aumentata preposta ai token degli attributi è:

seq_aug = [{t_CLS^{(i)}}{*i=1*}^{*N_CLS*}, {*t*{s̄_x}^{(i)}}{*i=1*}^{*L_pool*}, *t_z*, {*t_feat*^{*(j)*}}{j=1}^{D_feat}]

Dopo la trasformazione da parte del transformer, i parametri della distribuzione posteriore del bersaglio vengono estratti dalla porzione della sequenza corrispondente agli attributi mascherati m_trg^{(k)}:

q_φ(s_{y_k}^{(j)} | s_x, z, w; m_trg^{(k)}) = N(s_{y_k}^{(j)}; μ_sw_φ(h_aug)^{(j)}, Σ_sw_φ(h_aug)^{(j)}), j ∈ m_trg^{(k)}

Il modello predittivo. Il modulo p_θ(s_y^{(j)} | s_x, z; m_trg^{(k)}) opera interamente nello spazio latente, senza accesso agli osservabili w. Questa restrizione architetturale è il nucleo del paradigma JEPA: il predittore deve anticipare la struttura dello spazio di rappresentazione del bersaglio senza osservare direttamente i valori grezzi. Ciò si realizza attraverso token maschera apprendibili, inizializzati casualmente e aggiornati attraverso la discesa del gradiente, che fungono da "segnaposto" per gli attributi bersaglio:

h_ctx = Linear(s_x) + Linear(z) + e_pos^{ctx}

h_mask = MaskToken + Linear(z) + e_pos^{trg}

h_pred = Transformer(concat(h_ctx, h_mask))

p_θ(s_{y_k}^{(j)} | s_x, z; m_trg^{(k)}) = N(s_{y_k}^{(j)}; μ_sy_θ(h_pred)^{(j)}, Σ_sy_θ(h_pred)^{(j)})

L'addizione del termine Linear(z) sia alla sequenza contestuale sia ai token maschera consente all'informazione ausiliaria di modulare sia la rappresentazione del contesto sia quella del bersaglio predetto, implementando la dipendenza congiunta da s_x e z richiesta dalla formula generativa p_θ(s_y | s_x, z).

I decodificatori contestuale e bersaglio. Entrambi i decodificatori condividono la medesima architettura modulare, ma operano su sequenze latenti distinte. Per ciascun attributo j, esiste una testa decodificatrice dedicata u_θ^{(j)} la cui forma dipende dal tipo dell'attributo:

per j ∈ I_num: u_θ^{(j)} : ℝ^{D_hid} → ℝ¹, proiezione lineare

per j ∈ I_cat: u_θ^{(j)} : ℝ^{D_hid} → ℝ^{C_j}, proiezione lineare verso C_j logit

Le distribuzioni di ricostruzione per un osservabile generico $\pi \in \{x, w\}$ con rappresentazione latente s_π sono:

$$p_\theta(\pi^{(j)} | s_\pi^{(j)}) = N(\pi^{(j)}; u_\theta^{(j)}(s_\pi^{(j)}), \sigma^2), \; j \in I_{num}$$

$$p_\theta(\pi^{(j)} | s_\pi^{(j)}) = \text{Categorical}(\pi^{(j)}; \text{softmax}(u_\theta^{(j)}(s_\pi^{(j)}))), \; j \in I_{cat}$$

Il parametro σ^2 per la rumorosità di ricostruzione numerica è apprendibile a livello globale (uno per il percorso contestuale, uno per il percorso bersaglio), consentendo al modello di adattare il proprio livello di incertezza sulle ricostruzioni in funzione della complessità media degli attributi numerici nel dataset.

5.5 I termini di perdita adattati al dominio tabulare

Il passaggio dal framework Var-JEPA generale all'istanza tabulare richiede di ridefinire i cinque termini della funzione obiettivo in modo da tenere conto della struttura a maschera per attributo. Il termine di ricostruzione contestuale $\mathcal{L}_{rec}$ misura la qualità con cui il latente contestuale s_x recupera gli attributi mascherati x:

$$\mathcal{L}_{rec} = (1/M_{ctx}) \Sigma_{j \in m_{ctx}} [-\log p_\theta(x^{(j)} | s_x^{(j)})]$$

Il termine di generazione $\mathcal{L}_{gen}$ valuta invece la capacità del latente bersaglio s_w di ricostruire l'intera tupla w, non solo la porzione mascherata, operando dunque come una pressione verso rappresentazioni globalmente informative:

$$\mathcal{L}_{gen} = (1/D_{feat}) \Sigma_{j=1}^{D_{feat}} [-\log p_\theta(w^{(j)} | s_w^{(j)})]$$

I termini di divergenza KL per i latenti contestuale e ausiliario assumono forma chiusa grazie alla scelta di gaussiane diagonali con priori gaussiani standard. Per il latente contestuale:

$$\mathcal{L}_{KL_sx} = (1/M_{ctx}) \Sigma_{j \in m_{ctx}} KL(q_\varphi(s_x^{(j)} | x) \| N(0, I))$$

$$= (1/M_{ctx}) \Sigma_{j \in m_{ctx}} (1/2) \Sigma_{d=1}^{D_{hid}} [\Sigma_{sx_\varphi}^{(j,d)} + (\mu_{sx_\varphi}^{(j,d)})^2 - \log \Sigma_{sx_\varphi}^{(j,d)} - 1]$$

dove l'indice d percorre le dimensioni dello spazio latente D_{hid}. Analogamente per la variabile ausiliaria:

$$\mathcal{L}_{KL_z} = KL(q_\varphi(z | s_x) \| N(0, I)) = (1/2) \Sigma_{d=1}^{D_{hid}} [\Sigma_{z_\varphi}^{(d)} + (\mu_{z_\varphi}^{(d)})^2 - \log \Sigma_{z_\varphi}^{(d)} - 1]$$

Il termine più ricco di contenuto informativo è la divergenza KL tra il posteriore bersaglio e il modello predittivo, che in questo contesto tabulare viene mediata su tutti gli obiettivi K e su tutti gli attributi mascherati M_{trg}:

$$\mathcal{L}_{KL_sy} = (1/(K \cdot M_{trg})) \Sigma_{k=1}^{K} \Sigma_{j \in m_{trg}^{(k)}} KL(q_\varphi(s_{y_k}^{(j)} | s_x, z, w; m_{trg}^{(k)}) \| p_\theta(s_{y_k}^{(j)} | s_x, z; m_{trg}^{(k)}))$$

La forma chiusa di questa KL tra due gaussiane diagonali, con media μ_q e varianza Σ_q per il posteriore e μ_p e Σ_p per il predittivo, è:

$$KL(N(\mu_q, \Sigma_q) \| N(\mu_p, \Sigma_p)) = (1/2) \Sigma_{d=1}^{D_{hid}} [\Sigma_q^{(d)}/\Sigma_p^{(d)} + (\mu_p^{(d)} - \mu_q^{(d)})^2/\Sigma_p^{(d)} + \log(\Sigma_p^{(d)}/\Sigma_q^{(d)}) - 1]$$

Questa espressione, a differenza delle due precedenti, non si semplifica nella forma classica della divergenza verso la gaussiana standard, perché il priore condizionale $p_\theta(s_y|s_x, z)$ è apprendibile e dipende dai parametri θ. Ne consegue che il termine $\mathcal{L}_{KL_sy}$ porta un contributo non nullo anche quando le distribuzioni sono ben calibrate, e il suo minimizzatore non è trivialmente raggiungibile da un collasso della rappresentazione a costante.

La funzione obiettivo complessiva che l'algoritmo di addestramento minimizza è la combinazione pesata:

$$\mathcal{L} = \alpha_{rec} \cdot \mathcal{L}_{rec} + \alpha_{gen} \cdot \mathcal{L}_{gen} + \alpha_{KL_sx}(t) \cdot \mathcal{L}_{KL_sx} + \alpha_{KL_z}(t) \cdot \mathcal{L}_{KL_z} + \alpha_{KL_sy}(t) \cdot \mathcal{L}_{KL_sy}$$

dove i pesi delle divergenze KL seguono schedulazioni di annealing lineari:

$$\alpha_{KL_\cdot}(t) = \min(t / T_{anneal_\cdot}, 1) \cdot \alpha_{KL_\cdot,final}$$

Questa tecnica di annealing progressivo — introdotta da Bowman et al. (2016) nel contesto dei VAE per la generazione di frasi — è cruciale per prevenire il collasso posteriore nelle fasi iniziali dell'addestramento, quando il modello generativo non è ancora sufficientemente espressivo da fungere da priore condizionale informativo per il posteriore bersaglio. Incrementando gradualmente il peso delle penalità KL, si consente al modello di sviluppare prima buone rappresentazioni ricostruttive e poi di raffinarle attraverso la regolarizzazione variazionale.

5.6 I dati sperimentali e la valutazione a valle

La validazione empirica di Var-T-JEPA abbraccia cinque dataset tabulari reali — Adult (AD), Covertype (CO), Electricity (EL), Credit Card Default (CC), Bank Marketing (BM) — unitamente a una versione tabulare di MNIST (trattata come vettore di 784 pixel) e a un dataset sintetico controllato (SIM). La varietà dei domini — anagrafica censuaria, classificazione forestale, serie temporali di prezzo energetico, rischio creditizio, marketing diretto — garantisce una valutazione robusta non ristretta a un singolo contesto applicativo.

Per ciascun dataset, le rappresentazioni apprese da Var-T-JEPA e da T-JEPA vengono utilizzate come input per sei famiglie di predittori a valle: percettrone multistrato (MLP), DCNv2, ResNet, AutoInt, FT-Transformer e XGBoost. Il confronto con i medesimi predittori addestrati direttamente sulle feature grezze fornisce un riferimento per valutare se la preelaborazione non supervisionata aggiunge o sottrae informazione rispetto all'input originale.

I risultati documentati nella tavola comparativa rivelano una tendenza sistematica: Var-T-JEPA migliora in modo consistente rispetto a T-JEPA su tutti i dataset, mentre T-JEPA mostra su alcune combinazioni dataset-predittore una degradazione drastica delle

prestazioni riconducibile al collasso rappresentazionale. L'episodio più eclatante si osserva su Covertype con il predittore MLP: T-JEPA raggiunge un'accuratezza di appena 0,408 ± 0,231, contro lo 0,679 ± 0,029 di Var-T-JEPA, una differenza che segnala l'intera sequenza di rappresentazioni latenti degradata verso un attrattore costante. Su Electric con FT-Transformer, il collasso è persino più marcato: T-JEPA ottiene 0,317 ± 0,221, mentre Var-T-JEPA raggiunge 0,864 ± 0,009.

Il meccanismo che in Var-T-JEPA previene questo collasso è la regolarizzazione variazionale: la penalità KL esercita una pressione continua verso distribuzioni posteriori che non degenerano in delta di Dirac, mantenendo una varianza positiva per ogni dimensione dello spazio latente. In T-JEPA, la struttura deterministica con stop-gradient e aggiornamento EMA del codificatore bersaglio può condurre in certi regimi di addestramento a un attore di fuga stazionario in cui entrambi gli encoder collassano congiuntamente verso rappresentazioni quasi costanti, minimizzando banalmente la perdita di predizione.

Su dataset su cui entrambi i metodi non collassano, il vantaggio di Var-T-JEPA è più contenuto ma sistematico. Su Adult con DCNv2, il metodo variazionale ottiene 0,851 ± 0,002 contro 0,851 ± 0,002 di T-JEPA per accuratezza analoga ma con migliore stabilità; su Bank Marketing con MLP, 0,900 ± 0,002 contro 0,616 ± 0,371 di T-JEPA. Questa asimmetria nella varianza tra i due metodi è rivelatrice: Var-T-JEPA tende a essere non soltanto più accurato ma anche più stabile tra esecuzioni con seed differenti, attributo prezioso in contesti applicativi che richiedono riproducibilità.

5.7 La valutazione selettiva basata sull'incertezza

Una delle potenzialità specifiche del framework variazionale, che nessun approccio puramente deterministico può replicare, è la capacità di astenersi dalla predizione sui campioni per i quali l'incertezza stimata supera una soglia. In Var-T-JEPA, questa incertezza è computata aggregando le deviazioni standard del posteriore bersaglio $q_\varphi(s_y | s_x, z, w)$ per ogni campione di test, ottenendo così un punteggio scalare di confidenza.

Il protocollo di valutazione selettiva consiste nel scartare la frazione più incerta (10%, 20%, o 50%) dei campioni di test e misurare l'accuratezza sui restanti. I risultati documentati mostrano una curva rischio-copertura monotona decrescente: all'aumentare della frazione scartata, l'accuratezza sui campioni ritenuti cresce sistematicamente. Su Adult con XGBoost, la transizione da 0 al 50% di scarto eleva l'accuratezza da 0,855 a 0,928; su MNIST con il medesimo predittore, da 0,872 a 0,913. Questo comportamento, osservato in modo consistente su tutte le combinazioni dataset-predittore esaminate, dimostra che la varianza del posteriore latente porta informazione genuina sull'affidabilità della predizione a valle, e non è un mero artefatto del processo di stima variazionale.

La coerenza di questo andamento non è scontata: in un modello mal calibrato, la varianza del posteriore potrebbe non corrispondere alla difficoltà effettiva del campione, rendendo la selezione basata sull'incertezza non migliore di una selezione casuale. Il fatto che Var-T-JEPA produca curve rischio-copertura con pendenza negativa coerente su dataset così eterogenei suggerisce che la regolarizzazione variazionale attraverso i termini KL favorisce, come effetto collaterale benefico, una calibrazione dell'incertezza latente allineata con la difficoltà intrinseca dei campioni.

5.8 Connessioni con i metodi di apprendimento autosupervisionato tabulare esistenti: SCARF, VIME e TabNet

Var-T-JEPA non emerge nel vuoto: si inserisce in un filone di ricerca sull'apprendimento autosupervisionato per dati tabulari che negli ultimi anni ha prodotto diversi approcci di rilievo, ciascuno con una propria filosofia di mascheratura, aumentazione e obiettivo di apprendimento. Comprendere le relazioni formali tra Var-T-JEPA e questi metodi preesistenti chiarisce sia le innovazioni specifiche del contributo di Gögl e Yau sia le sue limitazioni rispetto alle soluzioni alternative.

SCARF (Bahri et al. 2022) è un metodo contrastivo per dati tabulari che genera viste aumentate sostituendo un sottoinsieme casuale di feature con valori campionati dalla distribuzione marginale empirica della feature corrispondente. La perdita SCARF è una InfoNCE contrastiva sulla coppia (x, x̂). VIME (Yoon et al. 2020) propone un framework a due pretext task: previsione della maschera e ricostruzione delle feature mascherate. L'obiettivo di ricostruzione di VIME opera nello spazio delle osservazioni, ed è soggetto alla critica di allocare capacità rappresentativa alla modellazione di variabilità a basso contenuto semantico. Var-T-JEPA supera questa limitazione operando nello spazio delle rappresentazioni latenti: invece di predire i valori grezzi delle feature mascherate, predice le loro rappresentazioni latenti attraverso il prior condizionale appreso p_θ(s_y | s_x, z).

TabNet (Arik e Pfenning, 2021) adotta meccanismi di attenzione selettiva per feature con un pretraining autosupervisionato basato su ricostruzione. Var-T-JEPA e TabNet condividono l'uso di meccanismi di attenzione multi-testa per elaborare le sequenze di token di feature, ma differiscono profondamente nell'obiettivo: TabNet ottimizza la ricostruzione nello spazio delle osservazioni, mentre Var-T-JEPA ottimizza l'ELBO variazionale nello spazio latente. I risultati di Var-T-JEPA su Covertype con predittore FT-Transformer (0.679 contro 0.408 di T-JEPA) suggeriscono che l'obiettivo variazionale fornisce un segnale di addestramento più efficace della ricostruzione deterministica per dataset con dipendenze inter-feature complesse.

5.9 Analisi della complessità computazionale e scalabilità di Var-T-JEPA

Il costo dominante nel forward pass di Var-T-JEPA è il meccanismo di auto-attenzione nel context encoder e nel target posterior, con complessità O((M_ctx)^2 * D_hid) e O((D_feat)^2 * D_hid) rispettivamente. Per D_feat grande (centinaia o migliaia di colonne, come nei dataset genomici o finanziari), il costo quadratico nella dimensione del token sequence può diventare proibitivo. Diverse strategie di riduzione della complessità sono disponibili nella letteratura dei transformer: l'attenzione lineare (Performer, Katharopoulos et al. 2020) riduce la complessità a O(D_feat * D_hid), mentre l'attenzione locale potrebbe essere adattata attraverso un ordinamento delle colonne basato sulla correlazione empirica.

Per la variabile ausiliaria z, il costo del pooling attenzionale e dell'MLP ausiliario è trascurabile rispetto al costo del transformer principale. Il costo del prior condizionale appreso p_θ(s_y | s_x, z) è O(M_trg * D_hid) — lineare nelle dimensioni del target mask, e controllabile attraverso la scelta di M_trg. I profili di memoria per i dataset tabulari standard (max D_feat ≈ 785 per MNIST tabulare) mostrano un footprint accettabile con batch size 256-512 su GPU standard, ma la scalabilità a D_feat > 10.000 richiederebbe l'adozione di tecniche di attenzione sub-quadratiche.

Capitolo 6 — Quantificazione dell'incertezza latente e connessione con VAE/β-VAE

6.1 L'incertezza come grandezza strutturale

Il problema della quantificazione dell'incertezza nelle reti neurali profonde ha ricevuto attenzione crescente negli ultimi anni, motivato dall'esigenza di sistemi che non soltanto producano predizioni, ma sappiano anche valutare il proprio grado di confidenza. La grande maggioranza degli approcci a questo problema si inserisce a posteriori su architetture altrimenti deterministiche: si pensi al dropout bayesiano, agli ensemble, o alle tecniche di calibrazione della confidenza basate su temperature scaling. Var-T-JEPA percorre invece una strada diversa: l'incertezza non è un'aggiunta posticcia, bensì una conseguenza strutturale della formulazione variazionale. Le varianze del posteriore q_φ(s_y|s_x, z, w) emergono naturalmente dall'ottimizzazione dell'ELBO come misura della dispersione della distribuzione posteriore sui latenti bersaglio, e portano con sé un'interpretazione probabilistica precisa.

Per comprendere a fondo il significato di questa incertezza occorre distinguere due fonti concettualmente distinte. La prima è l'incertezza aleatoria, insita nei dati stessi: due campioni con attributi contestuali identici ma bersagli diversi impongono per costruzione una distribuzione posteriore bersaglio dispersa, perché il modello generativo non può deterministicamente anticipare quale tra le possibili realizzazioni del bersaglio si materializzerà. La seconda è l'incertezza epistemica, che dipende dall'insufficienza informativa del contesto osservato: un campione in cui la maschera contestuale esclude gli attributi più predittivi del bersaglio produrrà un posteriore bersaglio con varianza elevata anche se il dataset non è intrinsecamente rumoroso. Nella pratica, le varianze del posteriore aggregano entrambe le sorgenti, e la loro interpretazione richiede cautela; tuttavia, come dimostrano i risultati sperimentali, anche questa misura aggregata porta informazione sufficiente per una selezione efficace dei campioni affidabili.

6.2 Stima dell'incertezza per campione e curve rischio-copertura

Dato un campione w di test, l'incertezza latente per campione è ottenuta aggregando le deviazioni standard marginali della distribuzione posteriore bersaglio q_φ(s_{y_k}^{(j)}|s_x, z, w; m_trg^{(k)}). Formalmente, per ciascun campione i si definisce un punteggio di

incertezza scalare u_i come funzione delle varianze del posteriore: nei dataset reali si impiega la media delle deviazioni standard su tutte le dimensioni latenti e su tutti gli attributi bersaglio mascherati; nei dataset semi-sintetici (MNIST e SIM), dove la struttura di incertezza è controllata, si preferisce il novantesimo percentile delle stesse deviazioni, al fine di catturare i comportamenti di coda — ovvero i campioni in cui anche soltanto alcune dimensioni latenti mostrano incertezza molto elevata.

La curva rischio-copertura (risk-coverage curve) è costruita ordinando i campioni di test per punteggio di incertezza crescente e calcolando l'errore del predittore a valle al variare della soglia di esclusione. Formalmente, dato un ordinamento u_{(1)} ≤ u_{(2)} ≤ … ≤ u_{(N)} dei campioni di test, la curva rischio-copertura è la mappa τ ↦ (1 − τ, Errore(campioni con u_i ≤ u_{(⌊τN⌋)})), dove τ ∈ [0,1] rappresenta la proporzione di campioni scartati e Errore è il tasso di errore del predittore a valle sui campioni ritenuti.

L'area sotto questa curva (AURC, Area Under the Risk-Coverage Curve) è una misura scalare della qualità dell'incertezza come strumento di selezione: valori bassi corrispondono a sistemi in cui l'esclusione dei campioni più incerti produce un decremento rapido e sostenuto dell'errore. Per MNIST, Var-T-JEPA raggiunge AURC = 0,123, che indica una curva discendente con pendenza pronunciata; per SIM, AURC = 0,209, un valore superiore che riflette la maggiore difficoltà intrinseca di questo dataset sintetico con struttura di ambiguità più complessa.

6.3 La correlazione tra incertezza latente e incertezza simulata

Nei dataset controllati, è possibile verificare direttamente in che misura l'incertezza latente stimata da Var-T-JEPA corrisponda all'incertezza effettivamente iniettata nel processo di generazione dei dati. Il dataset MNIST corrotto è costruito interpolando ogni immagine xi con un'immagine di rumore casuale ri tramite un coefficiente αi = clip(λui, 0, 1), dove ui ~ Uniform(0,1) è il punteggio di ambiguità simulata e λ = 0,75 è il fattore di scala massima del rumore:

x'_i = (1 − αi) · xi + αi · ri

Quanto più ui si avvicina a 1, tanto più l'immagine corrotta si discosta dall'originale e tanto più difficile diventa la classificazione. Il dataset SIM adotta una logica analoga: per ogni campione si campiona un punteggio ui ~ Uniform(0,1), che viene amplificato attraverso una mappa potenza u_i^{(amb)} = ui^γ con γ = 2 (così che la maggior parte dei campioni sia facile e soltanto una minoranza sia altamente ambigua), e impiegato per miscelare le feature numeriche di un campione verso il prototipo di una classe alternativa.

La correlazione di Spearman ϱ tra il punteggio di incertezza latente standardizzato di Var-T-JEPA e il punteggio di ambiguità simulata ui misura l'allineamento tra incertezza stimata e incertezza reale. Per MNIST, si osserva ϱ = 0,949, un valore eccezionalmente alto che attesta una quasi-biezione monotona tra le due grandezze: i campioni che il modello giudica più incerti corrispondono sistematicamente a quelli con maggior corruzione sintetica. Per SIM, ϱ = 0,640, un valore più moderato ma ancora altamente significativo, coerente con

la struttura di ambiguità più complessa che coinvolge sia attributi numerici sia categoriali e non si riduce a una semplice interpolazione lineare.

Il significato di questi risultati va oltre la semplice validazione empirica: essi mostrano che la varianza del posteriore variazionale non è un artefatto del processo di addestramento, ma cattura un segnale informativo genuino sulla struttura del dato. In un VAE mal addestrato o in un modello in cui il collasso posteriore sia parzialmente presente, le varianze tendono a convergere verso valori uniformi e uniformemente bassi, perdendo ogni relazione con la difficoltà del campione. La pressione della regolarizzazione KL — che mantiene il posteriore distante dalla degenrazione a delta di Dirac — è la condizione necessaria, ma non sufficiente, per preservare questa capacità informativa delle varianze.

6.4 Il rilevamento dei campioni ad alta ambiguità come problema di classificazione binaria

Un terzo angolo di analisi dell'incertezza consiste nel trattare il rilevamento dei campioni ad elevata ambiguità come un problema di classificazione binaria: un campione è considerato "ad alta ambiguità" se il suo punteggio simulato ui supera il settantacinquesimo percentile della distribuzione di ambiguità, e si verifica se il punteggio di incertezza latente è capace di discriminare questo sottoinsieme dal complemento.

La curva ROC costruita su questa classificazione binaria e la relativa area (AUC) forniscono una misura della capacità discriminativa dell'incertezza latente indipendente dalla scelta di una soglia specifica. Per MNIST, AUC = 0,986, corrispondente a una capacità discriminativa quasi perfetta: il modello identifica i campioni corrotti con alta affidabilità sulla base della sola varianza del posteriore latente, senza aver visto la variabile di ambiguità durante l'addestramento. Per SIM, AUC = 0,805, valore che conferma una discriminabilità buona anche in un contesto di ambiguità strutturalmente più ricca.

La combinazione di questi tre indicatori — AURC, correlazione di Spearman e AUC della ROC — costituisce un profilo di validazione dell'incertezza latente che va ben oltre la semplice misurazione dell'accuratezza del predittore a valle. Essa dimostra che il framework variazionale non si limita a produrre rappresentazioni migliori per compiti discriminativi, ma imprime sulla struttura delle distribuzioni posteriori latenti un'organizzazione allineata con la complessità intrinseca dei dati.

6.5 La connessione strutturale con il framework VAE accoppiato

Per comprendere il legame profondo tra Var-T-JEPA e la famiglia dei modelli a variabile latente variazionale, è utile tornare alla decomposizione del modello generativo derivata nel capitolo 3. La struttura del modello di Var-JEPA:

$$p_\theta(x, y, s_x, z, s_y) = p(s_x) \cdot p(z) \cdot p_\theta(x \mid s_x) \cdot p_\theta(s_y \mid s_x, z) \cdot p_\theta(y \mid s_y)$$

è formalmente equivalente a quella di un VAE accoppiato (coupled VAE) nella terminologia di Hao & Shafto (2023). In un VAE standard, un singolo osservabile x è generato a partire da un singolo latente s attraverso la catena $p(s) \cdot p_\theta(x|s)$; le distribuzioni posterori e priori sono gaussiane e l'ELBO risultante ha la forma classica:

$$\mathrm{ELBO_VAE} = \mathbb{E}_{q_\varphi(s|x)}[\log p_\theta(x|s)] - \mathrm{KL}(q_\varphi(s|x) \,\|\, p(s))$$

Nel VAE accoppiato, due osservabili x e y sono generati da due latenti correlati s_x e s_y, con la dipendenza tra i latenti mediata da un priore condizionale $p_\theta(s_y|s_x)$. Var-JEPA introduce una terza variabile latente ausiliaria z, che funge da sorgente di variabilità non spiegabile dal contesto, e fattorizza il posteriore variazionale come $q_\varphi(s_x|x) \cdot q_\varphi(z|s_x) \cdot q_\varphi(s_y|s_x, z, y)$.

Il predittore JEPA $p_\theta(s_y|s_x, z)$ emerge dunque non come un modulo esterno aggiunto all'architettura, bensì come il priore condizionale latente del modello generativo accoppiato. Questa identificazione è di profonda rilevanza concettuale: il predittore non "predice" in senso discriminativo (da ingresso a uscita senza struttura probabilistica), ma agisce come distribuzione generativa condizionale che specifica quali realizzazioni del latente bersaglio sono compatibili con il latente contestuale e la variabile ausiliaria. La penalità KL del posteriore bersaglio verso questo priore condizionale è la formalizzazione dell'obiettivo di predizione come regolarizzazione di Kullback-Leibler, non come minimizzazione di una distanza punto a punto.

6.6 La decomposizione ELBO-surgery e il collo di bottiglia informativo

La tecnica nota come "ELBO surgery", introdotta da Hoffman & Johnson (2016), consente di scomporre la divergenza KL aggregata di un posteriore variazionale in due termini separati con interpretazioni distinte. Per i latenti con priore fisso standard — s_x e z in Var-JEPA — la divergenza KL per campione ammette la decomposizione:

$$\mathbb{E}_x[\mathrm{KL}(q_\varphi(s_x|x) \,\|\, N(0, I))] = \mathrm{KL}(q_\varphi(s_x) \,\|\, N(0, I)) + I_{q_\varphi}(x; s_x)$$

dove $q_\varphi(s_x) = \int q_\varphi(s_x|x)\, p(x)\, dx$ è il posteriore aggregato (la distribuzione marginale del latente sulla distribuzione dei dati), $\mathrm{KL}(q_\varphi(s_x) \,\|\, N(0, I))$ misura la distanza tra questa marginale aggregata e la gaussiana standard, e $I_{q_\varphi}(x; s_x)$ è l'informazione mutua tra il dato di ingresso e il latente sotto la distribuzione variazionale.

Questa decomposizione svela un duplice ruolo della penalità KL nell'ELBO: da un lato pressa la distribuzione aggregata dei latenti verso un'isotropia gaussiana (il primo termine); dall'altro agisce come collo di bottiglia informativo, penalizzando la quantità di informazione che il latente trattiene dall'ingresso (il secondo termine). Il bilanciamento tra questi due effetti è governato dalla stessa penalità KL, senza necessità di iperparametri aggiuntivi — a meno di scegliere di pesare diversamente i due contributi, come avviene nel β-VAE.

Per il latente bersaglio s_y, la decomposizione analoga non si applica in modo diretto: la penalità KL del posteriore bersaglio in Var-JEPA è verso il priore condizionale apprendibile $p_\theta(s_y|s_x, z)$, non verso una gaussiana standard fissa. Ne consegue che la distribuzione

marginale aggregata $q_\varphi(s_y)$ non converge necessariamente verso $N(0, I)$, come confermato dai dati sperimentali dello studio di simulazione: la divergenza KL aggregata di $q_\varphi(s_y)$ rispetto a $N(0, I)$ è sistematicamente elevata (circa 3,5 nello scenario standard), mentre quella di $q_\varphi(s_x)$ è contenuta (circa 0,11), prossima a quanto si otterrebbe con una regularizzazione SIGReg esplicita.

Questa asimmetria è teoricamente corretta e computazionalmente benefica: s_x deve rimanere vicino alla gaussiana standard perché campionato da un priore gaussiano durante la generazione da contesto; s_y deve invece adattarsi al priore condizionale $p_\theta(s_y | s_x, z)$, che a sua volta si allinea con i dati reali attraverso l'addestramento.

6.7 La connessione con il β-VAE attraverso la pesatura dell'ELBO

Il β-VAE di Higgins et al. (2017) è una variante del VAE classico in cui la penalità KL è moltiplicata da un iperparametro $\beta > 1$:

$$\mathcal{L}_{\beta\text{-VAE}} = -\mathbb{E}_{q_\varphi(s|x)}[\log p_\theta(x|s)] + \beta \cdot \mathrm{KL}(q_\varphi(s|x) \,\|\, p(s))$$

Il parametro β controlla il compromesso tra fedeltà della ricostruzione e isotropia del posteriore aggregato: valori elevati di β producono rappresentazioni più disentangled — dove diverse dimensioni del latente corrispondono a fattori di variazione indipendenti nei dati — a costo di una ricostruzione meno accurata.

La funzione obiettivo di Var-JEPA con pesi distinti α_{rec}, α_{gen}, α_{KL_sx}, α_{KL_z}, α_{KL_sy} è formalmente analoga a una versione multi-termine del β-VAE in cui ciascun contributo è pesato indipendentemente. Il peso α_{KL_sx} governa l'isotropia di $q_\varphi(s_x)$ e il collo di bottiglia informativo per il contesto; α_{KL_z} governa la complessità della variabile ausiliaria; α_{KL_sy} governa la pressione del posteriore bersaglio verso il priore condizionale predittivo. L'analisi di sensibilità condotta su Adult mostra che le prestazioni a valle sono robuste a variazioni di un ordine di grandezza di questi pesi: spostare α_{KL_sx} da 10^{-4} a 10^{-3} o α_{KL_sy} da 10^{-5} a 10^{-4} non produce degradazioni significative dell'accuratezza del predittore a valle, anche se il profilo delle distribuzioni latenti varia.

Questa robustezza è al tempo stesso un vantaggio pratico — riducendo il costo del tuning degli iperparametri — e un indicatore teorico del fatto che l'ELBO non è sensibile ai pesi precisi entro un certo intorno dei valori nominali. La regolarizzazione variazionale opera su scale di pressione sufficientemente diverse dai termini ricostruttivi che piccole variazioni relative dei pesi KL non alterano significativamente il bilanciamento complessivo dell'ottimizzazione.

6.8 SIGReg come regolarizzazione della distribuzione aggregata e suo ruolo complementare

Un ulteriore filo di connessione con la letteratura sulla normalizzazione delle rappresentazioni è offerto dalla Sketched Isotropic Gaussian Regularization (SIGReg),

proposta da Balestriero & LeCun (2025) nel contesto di LeJEPA. SIGReg applica un test statistico univariato T (tipicamente il test di Epps-Pulley, 1983) alle proiezioni degli embedding lungo direzioni casuali unitarie A = {a₁, …, a_M}:

SIGReg(A, {e_n}{*n=1*}^{*N*}*)* = *(1/|A|)* Σ{a ∈ A} T({a^⊤ e_n}_{n=1}^{N})

dove T misura la discrepanza tra la distribuzione empirica delle proiezioni e la gaussiana standard. L'obiettivo è forzare la distribuzione aggregata degli embedding verso N(0, I) attraverso un test proiettivo scalabile, evitando il costo computazionale della stima della covarianza in alta dimensione.

Dallo studio di simulazione emerge che i termini KL per campione della penalità ELBO producono, come effetto aggregato, una distribuzione di q_φ(s_x) già prossima alla gaussiana standard: la distanza KL aggregata KL(q_φ(s_x) ‖ N(0, I)) si attesta su valori di circa 0,11 senza alcuna SIGReg, comparabili a quelli ottenibili con λ_sx = 10 di regolarizzazione SIGReg esplicita. Questo risultato conferma la previsione teorica della decomposizione ELBO-surgery: minimizzare la KL per campione verso la gaussiana standard implica implicitamente minimizzare la distanza della distribuzione aggregata dalla stessa gaussiana standard, pur senza ottimizzare direttamente questa quantità.

L'aggiunta di SIGReg all'ELBO (varianti B, C, D dello studio di simulazione) non migliora le prestazioni della sonda lineare su s_x, che rimane stabile a circa 0,996 in tutte le configurazioni. Migliora invece la isotropia aggregata di q_φ(s_y) quando applicata direttamente a quel latente (variante C: KL_agg da 3,53 a 1,999), segnalando che per il latente bersaglio — governato da un priore condizionale apprendibile e non da un priore fisso — la regolarizzazione aggregata esplicita fornisce un contributo genuino che la penalità KL per campione non replica automaticamente.

La rimozione della ricostruzione (variante G, α_rec = α_gen = 0) produce invece un collasso rappresentazionale evidente: l'accuratezza della sonda scende a 0,571 da 0,996. Questo collasso si manifesta nonostante la regolarizzazione KL sia ancora presente, il che indica che la ricostruzione svolge un ruolo fondamentale nel vincolare il posteriore a codificare informazione pertinente al dato osservabile, e non semplicemente nel conformarsi al priore. L'aggiunta di SIGReg in assenza di ricostruzione (variante H) recupera parzialmente le prestazioni (0,834), ma non raggiunge il livello ottenuto con l'ELBO completo, confermando che i termini ricostruttivi e la regolarizzazione variazionale sono complementari e non sostituibili l'uno con l'altro.

6.9 Interpretazione unitaria: JEPA come caso deterministico di modello latente variazionale

La riflessione condotta in questo capitolo converge verso una tesi di carattere generale: la dicotomia tra apprendimento predittivo alla JEPA e modellazione generativa variazionale è in larga misura una questione di parametrizzazione e non di struttura. Il JEPA deterministico è un punto particolare nello spazio di progettazione dei modelli a variabile latente, ottenuto portando le varianze del posteriore a zero (limite deterministico) e sostituendo la

regolarizzazione esplicita con euristiche architetturali — stop-gradient, EMA — che ne replicano gli effetti in modo approssimato.

Var-JEPA rende esplicita questa struttura latente e, così facendo, acquista due capacità che il JEPA deterministico non può avere: una quantificazione formalmente fondata dell'incertezza, e una prevenzione del collasso rappresentazionale che non dipende da meccanismi ad hoc ma emerge dall'ottimizzazione di un unico obiettivo variazionale coerente. I risultati empirici su dati tabulari — sia in termini di accuratezza a valle sia in termini di calibrazione dell'incertezza — avvalorano questa interpretazione e suggeriscono che la dimensione probabilistica del framework non è un lusso teorico, ma un ingrediente funzionale per la robustezza e la generalizzazione del modello.

Nel capitolo successivo si abbandonerà temporaneamente il filo variazionale per muovere verso il secondo asse tematico della monografia parallela: l'integrazione dell'obiettivo JEPA nell'addestramento dei modelli linguistici di grande scala, attraverso il contributo di Huang, LeCun e Balestriero (2025) su LLM-JEPA.

6.10 Decomposizione formale dell'incertezza epistemica e aleatoria nella previsione variazionale

La distinzione tra incertezza epistemica (dovuta a mancanza di dati o di esperienza) e incertezza aleatoria (dovuta a variabilità intrinseca irriducibile del fenomeno) è fondamentale per l'uso pratico delle stime di incertezza in sistemi di supporto alle decisioni. Var-T-JEPA produce un'incertezza aggregata --- la varianza del posteriore q_φ(s_y | s_x, z, y) --- che mescola entrambe le sorgenti senza distinguerle esplicitamente. Formalizzare questa distinzione permette di identificare limiti dell'approccio attuale e direzioni per architetture future più informative.

Nella decomposizione bayesiana canonica, l'incertezza predittiva totale si scompone attraverso la legge della varianza totale: Var[s_y | x] = E_θ[Var[s_y | x, θ]] + Var_θ[E[s_y | x, θ]], dove il primo termine è l'incertezza aleatoria (varianza della distribuzione predittiva con parametri fissati) e il secondo è l'incertezza epistemica (varianza dell'aspettazione predittiva rispetto all'incertezza sui parametri). Per Var-T-JEPA con parametri deterministici (θ, φ) appresi per massima verosimiglianza variazionale, il secondo termine è nullo per definizione: il modello non ha incertezza epistemica sui propri parametri. Tutta l'incertezza residua è quindi aleatoria --- riflette la variabilità intrinseca della distribuzione del target condizionato al contesto.

Tuttavia, nella pratica, la varianza del posteriore q_φ(s_y | s_x, z, y) porta anche un segnale di out-of-distribution (OOD) che è essenzialmente epistemico: un campione di test con feature contestuali x mai viste durante l'addestramento produrrà un'incertezza elevata non perché il fenomeno sia intrinsecamente aleatorio, ma perché il modello non ha esperienza per fare una previsione affidabile. Questo segnale OOD emerge empiricamente nei risultati di Gögl e Yau (correlazione $\varrho = 0.949$ tra incertezza latente e corruzione simulata su MNIST), ma non è distinguibile a livello formale dall'incertezza aleatoria pura.

Una separazione più precisa è realizzabile attraverso tecniche di deep ensemble (Lakshminarayanan et al. 2017) o di deep Bayesian learning applicato al modello Var-T-JEPA. Addestrando K copie indipendenti del modello con diverse inizializzazioni e/o diverse sotto-campionature del training set, l'incertezza epistemica può essere stimata come la varianza delle K medie posteriori E_{φ_k}[s_y | s_x, z, y], mentre l'incertezza aleatoria è la media delle K varianze posteriori Var_{φ_k}[s_y | s_x, z, y]. Per K = 5 (numero tipico negli ensemble nella letteratura), questa decomposizione richiede 5× il costo computazionale di addestramento, ma fornisce informazioni molto più ricche sulla natura dell'incertezza --- cruciale in applicazioni ad alta responsabilità come la diagnosi medica automatizzata.

6.11 Metriche di calibrazione dell'incertezza: Expected Calibration Error e reliability diagrams

La correlazione di Spearman tra incertezza latente e difficoltà dei campioni, riportata da Gögl e Yau, misura la capacità ordinale dell'incertezza di discriminare campioni facili da campioni difficili. Questa metrica è informativa per il problema della selezione selettiva (decidere quali campioni scartare), ma non misura direttamente se il livello assoluto di confidenza dichiarato dal modello corrisponda alla probabilità effettiva di errore --- proprietà nota come calibrazione.

L'Expected Calibration Error (ECE, Naeini et al. 2015) misura la calibrazione di un classificatore discreto binando le predizioni per livello di confidenza e confrontando la confidenza media di ciascun bin con l'accuratezza media nello stesso bin: ECE = Σ_{m=1}^M (|B_m|/n) · |acc(B_m) − conf(B_m)|, dove B_m è il m-esimo bin, acc(B_m) l'accuratezza nel bin e conf(B_m) la confidenza media. Per un classificatore perfettamente calibrato, ECE = 0: ogni volta che il modello dichiara confidenza 80%, è corretto nell'80% dei casi. Per Var-T-JEPA, l'analogo dell'ECE richiederebbe di trasformare la varianza del posteriore latente in un'estimate di probabilità di errore downstream, operazione non immediata dato che la varianza è nello spazio latente e l'errore è nello spazio delle predizioni a valle.

Un approccio praticabile per valutare la calibrazione di Var-T-JEPA è il seguente. Per ciascun campione i, si calcola il punteggio di incertezza u_i (media delle deviazioni standard posteriori). Si definiscono bin di uguale dimensione rispetto a u_i, e si misura, per ciascun bin, l'errore medio del predittore a valle (ad esempio tasso di classificazione errata con una sonda lineare). I reliability diagram così ottenuti mostrano la relazione tra incertezza dichiarata e errore effettivo. Un modello perfettamente calibrato produrebbe una relazione monotona crescente: campioni con incertezza più alta dovrebbero avere un tasso di errore più alto. La corrispondenza empirica osservata nelle curve rischio-copertura di Gögl e Yau (dove scartare i campioni più incerti migliora sistematicamente l'accuratezza) è coerente con una buona calibrazione ordinale, ma non è sufficiente per affermare la calibrazione assoluta senza un'analisi esplicita dei reliability diagram.

Capitolo 7 — Perché i LLM possono beneficiare di un obiettivo JEPA

7.1 La frattura tra visione e linguaggio nella letteratura sull'apprendimento delle rappresentazioni

Negli ultimi anni il campo dell'apprendimento delle rappresentazioni si è progressivamente polarizzato attorno a due paradigmi distinti, che corrispondono a strategie fondamentalmente differenti per estrarre struttura dai dati non etichettati. Da un lato, i metodi generativi o basati sulla ricostruzione adottano come obiettivo la riproduzione fedele dell'ingresso nello spazio osservabile: si addestrano modelli a predire pixel, token o campioni audio a partire da versioni parziali o rumorose dello stesso segnale, e la qualità delle rappresentazioni interne viene misurata implicitamente dalla capacità di questa ricostruzione. Dall'altro, le architetture a embedding congiunto predittivo — le JEPA — rinunciano alla ricostruzione nello spazio osservabile e lavorano interamente nello spazio delle rappresentazioni: il modello apprende a predire l'embedding di una vista del dato a partire dall'embedding di un'altra vista, evitando di dover modellare la distribuzione completa dei pixel o dei token.

Questa dicotomia ha trovato risoluzione empirica nel dominio visivo: risultati teorici e sperimentali convergenti hanno mostrato che le JEPA offrono vantaggi significativi rispetto ai metodi ricostruttivi per i compiti di percezione e ragionamento visivo. L'intuizione è che la ricostruzione impegna la capacità rappresentazionale del modello a codificare dettagli di basso livello — texture, rumore, variazioni irrilevanti dell'illuminazione — che non contribuiscono alla comprensione semantica, mentre la predizione in spazio latente spinge il modello a catturare relazioni strutturali tra porzioni del dato, apprendendo rappresentazioni invarianti rispetto alle informazioni irrilevanti alla predizione.

Nel dominio del linguaggio naturale, questa transizione non si è ancora verificata. I modelli linguistici di grande scala continuano a essere addestrati quasi esclusivamente attraverso obiettivi ricostruttivi — la predizione autoregressiva del token successivo o il masked language modeling — e vengono valutati principalmente sulla base delle loro capacità generative nello spazio degli ingressi: produrre testo coerente, rispondere a domande, generare codice. La ragione di questa persistenza non è esclusivamente conservatrice: definire "viste" per il linguaggio naturale in modo analogo a come le maschere spaziali definiscono viste per le immagini è un problema concettuale non banale, e la mancanza di una soluzione chiara ha finora impedito l'adozione su larga scala degli obiettivi JEPA nel dominio testuale.

7.2 Il problema della definizione delle viste in ambito linguistico

Nel paradigma visivo, due viste dello stesso dato sono ottenute applicando trasformazioni che preservano la semantica ma alterano la superfice percettiva: ritagli di un'immagine in

posizioni diverse, applicazione di augmentation cromatiche, mascheratura di patch spaziali. Queste trasformazioni sono ben definite, geometricamente interpretabili e facilmente campionabili. Il modello addestrato a predire l'embedding di una vista dall'embedding dell'altra deve imparare a estrarre le componenti semantiche condivise tra le due, che corrispondono per costruzione al contenuto informativo rilevante dell'immagine.

Per il linguaggio naturale non esiste un equivalente diretto di questo meccanismo di augmentation. Due trascrizioni leggermente diverse dello stesso enunciato non sono "viste" nel senso JEPA: non mantengono necessariamente le stesse relazioni semantiche, e la loro generazione richiede capacità parafrastica che si vorrebbe appunto acquisire, non presupporre. Il dropout applicato ai token, utilizzato in approcci come SimCSE per ottenere viste pseudo-diverse della medesima frase, produce diversità troppo superficiale per motivare l'apprendimento di rappresentazioni semanticamente profonde.

La soluzione proposta da Huang, LeCun e Balestriero (2025) è concettualmente elegante nella sua semplicità: invece di costruire viste attraverso trasformazioni artificiali, si sfrutta la struttura naturale di certi dataset in cui due rappresentazioni complementari dello stesso contenuto sono già presenti come coppie di esempi distinte. L'intuizione centrale è che alcune tipologie di dati esibiscono per natura una dualità di prospettiva — una formulazione in linguaggio naturale e una formulazione formale o procedurale dello stesso oggetto conoscitivo — e che questa dualità costituisce esattamente la struttura necessaria per l'obiettivo JEPA.

7.3 Le coppie naturalmente duali come viste del medesimo substrato conoscitivo

La casistica dei dataset con struttura a doppia vista è più ampia di quanto possa sembrare a prima riflessione. I tre esempi fondamentali documentati nel contributo di Huang et al. illustrano la varietà di questo pattern.

Il primo e più diretto è la traduzione dal linguaggio naturale a espressioni regolari (NL→RegEx), incarnata dal dataset NL-RX-SYNTH e dalla sua variante NL-RX-TURK. Ogni campione è costituito da una descrizione verbale dell'insieme di stringhe da catturare — ad esempio "linee che contengono la stringa 'dog' seguita da un numero, tre o più volte" — e dalla corrispondente espressione regolare, come `((dog.*[0-9].*)3,)`. Le due componenti sono rappresentazioni isomorfiche del medesimo vincolo semantico, espresse rispettivamente nel linguaggio naturale e in un linguaggio formale di pattern matching. Nessuna delle due è riducibile all'altra per semplice trasformazione sintattica: la traduzione richiede comprensione strutturale del significato.

Il secondo caso è la traduzione dal linguaggio naturale a SQL (NL→SQL), rappresentata dal dataset Spider. Ogni esempio associa una descrizione verbale di una interrogazione su una base di dati — come "quante teste di dipartimento hanno più di 56 anni?" — alla corrispondente query SQL: `SELECT count(*) FROM head WHERE age > 56`. Anche in questo caso, la coppia costituisce due proiezioni della medesima intenzione informativa, una rivolta all'interlocutore umano e l'altra alla macchina.

Il terzo pattern, più ricco di sfumature, è quello della risoluzione di problemi matematici attraverso programmi o sequenze di ragionamento (GSM8K): un problema espresso in linguaggio naturale e la sua soluzione sotto forma di catena di calcoli e ragionamenti intermedi. In questo contesto, la "vista" testuale e la "vista" procedurale non sono equivalenti per contenuto ma complementari per struttura: la prima veicola il contesto e i vincoli del problema, la seconda il percorso di soluzione.

Un quarto pattern, citato nel contributo ma non esplorato in modo esaustivo per ragioni computazionali, è quello delle coppie (descrizione di issue, diff di codice) nei repository software. Un'issue GitHub descrive in linguaggio naturale un difetto o una funzionalità richiesta; il corrispondente commit contiene la modifica al codice che risolve o implementa quanto descritto. Le due componenti sono viste distinte della medesima trasformazione funzionale del software, espresse rispettivamente nel registro comunicativo e in quello operativo.

7.4 La struttura matematica dell'obiettivo JEPA per LLM

Il punto di partenza è la funzione di perdita standard dei LLM, che per un task generico su una sequenza di lunghezza L è la cross-entropia autoregressiva:

$$\mathcal{L}_{LLM}(Text_{1:L-1}, Text_L) = XEnt(Classifier(Enc(Text_{1:L-1})), Text_L)$$

dove Enc(·) è il codificatore del modello linguistico, Classifier(·) è la testa di proiezione verso il vocabolario, e XEnt(·,·) è la cross-entropia tra la distribuzione predetta e il token di riferimento. Nella pratica del preaddestramento autoregressivo, questa perdita viene accumulata su tutti i passi temporali attraverso la causalità del meccanismo di attenzione:

$$\mathcal{L}_{LLM} = \Sigma_{\ell=2}^{L} \mathcal{L}_{LLM}(Text_{1:\ell-1}, Text_\ell)$$

L'obiettivo JEPA per LLM viene aggiunto come termine supplementare che agisce nello spazio degli embedding: il predittore Pred(·) produce un vettore di embedding a partire dall'embedding dell'ingresso testuale Enc(Text), e questo vettore viene confrontato con l'embedding indipendentemente calcolato dell'ingresso di codice Enc(Code) attraverso una metrica d(·,·):

$$\mathcal{L}_{LLM-JEPA} = \Sigma_{\ell=2}^{L} \mathcal{L}_{LLM}(Text_{1:\ell-1}, Text_\ell) + \lambda \times d(Pred(Enc(Text)), Enc(Code))$$

dove $\lambda \geq 0$ è l'iperparametro di bilanciamento tra il contributo generativo e quello di predizione in spazio latente. La struttura di questa perdita composita è deliberatamente additiva e non sostitutiva: il termine generativo $\mathcal{L}_{LLM}$ non viene soppiantato dall'obiettivo JEPA, ma preservato nella sua interezza. Questa scelta ha implicazioni fondamentali che verranno esaminate in seguito.

7.5 L'evidenza empirica dell'ortogonalità dei due obiettivi

Una domanda cruciale, preliminare a qualunque argomento sull'utilità dell'obiettivo JEPA per i LLM, è se il termine aggiuntivo sia davvero necessario o se la minimizzazione di $\mathcal{L}$_LLM non implichi già implicitamente la minimizzazione della perdita di predizione JEPA $\mathcal{L}$_JEPA = d(Pred(Enc(Text)), Enc(Code)). Se così fosse, l'aggiunta dell'obiettivo JEPA non porterebbe nuovo segnale di addestramento e il miglioramento delle prestazioni dovrebbe essere attribuito ad altri fattori.

La risposta è fornita da un esperimento controllato su Llama-3.2-1B-Instruct e NL-RX-SYNTH con due configurazioni di addestramento: nella prima, il gradiente del termine JEPA non viene propagato (il termine è monitorato ma non ottimizzato); nella seconda, il termine JEPA contribuisce al gradiente insieme al termine LLM standard. I risultati mostrano con chiarezza che minimizzare $\mathcal{L}$_LLM non minimizza $\mathcal{L}$_JEPA: la perdita di predizione JEPA rimane pressoché costante nel corso dell'addestramento con il solo obiettivo generativo, mentre decresce in modo sostenuto con l'obiettivo composito. I due termini occupano regioni distinte dello spazio dei parametri rispetto alle quali il modello è sensibile: il gradiente di $\mathcal{L}$_LLM non porta informazione nelle direzioni che permettono di avvicinare Enc(Text) a Pred(Enc(Code)), perché l'obiettivo autoregressivo non impone alcun vincolo esplicito sulla struttura geometrica degli embedding degli ingressi testuali rispetto agli embedding degli ingressi formali corrispondenti.

Al contempo, la perdita autoregressiva $\mathcal{L}$_LLM nella configurazione con obiettivo composito non mostra incremento rispetto alla configurazione con il solo obiettivo generativo: le traiettorie di discesa della cross-entropia sono sovrapposte nelle due condizioni, a indicare che l'aggiunta del termine JEPA non produce degradazione della capacità generativa del modello. Questo risultato — già osservato empiricamente nel dominio visivo da Balestriero e LeCun (2024) per gli autoencoder — suggerisce che le informazioni codificate nelle rappresentazioni per la predizione autoregressiva e quelle necessarie per la predizione in spazio latente non si contendono le stesse direzioni nello spazio dei parametri, ma si integrano in modo quasi ortogonale. L'obiettivo JEPA aggiunge struttura allo spazio latente senza alterare la traiettoria dell'ottimizzazione generativa.

7.8 Teoria delle viste ottimali per il linguaggio: quando due sequenze costituiscono viste genuine

La nozione di "viste" in JEPA --- due rappresentazioni della stessa conoscenza sottostante che differiscono nella forma ma concordano nella sostanza --- è intuitivamente chiara per il dominio visivo (due ritagli dello stesso oggetto) ma richiede una formalizzazione più attenta per il linguaggio naturale. Quando due sequenze linguistiche y_1 e y_2 costituiscono viste genuine dello stesso contenuto? La risposta a questa domanda determina quali dataset sono adatti per LLM-JEPA e quali pattern di costruzione delle viste possono essere automatizzati.

Una formalizzazione possibile si basa sulla nozione di equivalenza semantica condizionale. Due sequenze y_1 e y_2 sono viste dello stesso contenuto sottostante s se e solo se esistono codificatori E_1 e E_2 tali che $E_1(y_1) \approx E_2(y_2) \approx s$, dove s è una rappresentazione latente condivisa. Questa definizione è circolare se E_1 e E_2 devono essere appresi --- è precisamente ciò che LLM-JEPA cerca di apprendere --- ma permette di formulare una condizione sufficiente: y_1 e y_2 sono viste genuine se le loro rispettive trascrizioni semantiche formali sono equivalenti. Per le coppie (descrizione NL, espressione regolare), questa equivalenza semantica formale è verificabile algoritmicamente: due espressioni regolari sono semanticamente equivalenti se riconoscono lo stesso linguaggio formale, e una descrizione in NL è semanticamente equivalente a un'espressione regolare se quella espressione riconosce esattamente l'insieme descritto dalla descrizione.

Per dataset meno strutturati, come le parafrasi del dataset cestwc/paraphrase o i completamenti di HellaSwag, la nozione di equivalenza semantica è più sfumata. Due parafrasi possono differire per implicature pragmatiche, presupposizioni contestuali o struttura informativa (topic/focus), pur concordando sui valori di verità delle proposizioni principali. Questo suggerisce che "viste genuine" per il linguaggio dovrebbero soddisfare una condizione di invarianza sotto trasformazioni che preservano le condizioni di verità ma non necessariamente la struttura retorica o pragmatica. La definizione formale di questa invarianza richiederebbe strumenti dalla semantica formale computazionale --- teoria dei tipi intensionali, logica delle proprietà, teorie della grounded semantics --- che vanno al di là dello scope del presente paper ma costituiscono una direzione di ricerca teoricamente fondamentale per lo sviluppo di LLM-JEPA su corpus generici.

Un risultato pratico immediatamente derivabile da questa analisi riguarda la qualità del segnale di addestramento in funzione del grado di complementarità delle viste. Huang, LeCun e Balestriero documentano che i miglioramenti di LLM-JEPA sono maggiori su NL-RX-SYNTH (dove testo e codice sono semanticamente complementari e formalmente equivalenti) che su GSM8K (dove testo e soluzione numerica sono asimmetrici) e HellaSwag (dove contesto e completamento sono sequenzialmente correlati ma non equivalenti). Questa gerarchia è coerente con la teoria: viste più genuinamente equivalenti producono un segnale di allineamento latente più informativo, tradotto in un miglioramento maggiore delle prestazioni downstream. La costruzione di dataset con viste ottimali per il pre-addestramento di LLM-JEPA su corpus generici è dunque un problema aperto con importanti conseguenze pratiche.

7.9 Estensione multilingue e cross-lingual di LLM-JEPA

Una direzione di estensione particolarmente promettente di LLM-JEPA, non esplorata nel paper originale di Huang et al. (2025), è l'applicazione dell'obiettivo JEPA a coppie di testi in lingue diverse come viste dello stesso contenuto. Questa estensione è teoricamente motivata dalla struttura stessa della traduzione: una frase in inglese e la sua traduzione in italiano sono due rappresentazioni della stessa proposizione, espresse in

sistemi linguistici diversi. Se LLM-JEPA può apprendere a allineare le rappresentazioni di testo e codice, potrebbe analogamente allineare le rappresentazioni di testi in lingue diverse, producendo un embedding multilingue strutturato.

La connessione con il filone della ricerca cross-lingual è diretta. I modelli multilingui come mBERT (Devlin et al. 2019) e XLM-R (Conneau et al. 2020) apprendono rappresentazioni condivise per più lingue attraverso il pre-addestramento congiunto su corpus monolingui con architettura shared. Questi modelli mostrano un'emergenza spontanea di rappresentazioni cross-lingual --- la capacità di generalizzare tra lingue non vista durante il training di compiti specifici (zero-shot cross-lingual transfer) --- che è parzialmente spiegata dall'allineamento dei sottospazi linguistici attraverso il vocabolario condiviso (caratteri e sottotesti comuni a più lingue). LLM-JEPA con viste multilingue potrebbe amplificare questo allineamento aggiungendo un vincolo esplicito di prossimità nell'embedding space tra rappresentazioni di testi equivalenti in lingue diverse, potenzialmente migliorando la qualità del trasferimento cross-lingual.

Formalmente, la perdita LLM-JEPA multilingue assumerebbe la forma: $\mathcal{L}_ML\text{-}JEPA = \Sigma_{\ell=2}^{L} \mathcal{L}_LLM(Text_L1, Text_L1) + \lambda \cdot d(Pred(Enc(Text_L1)), Enc(Text_L2))$, dove Text_L1 e Text_L2 sono la stessa proposizione in due lingue diverse L1 e L2. La scelta di d come similarità coseno --- che normalizza le norme dei vettori --- è particolarmente appropriata nel contesto multilingue, poiché le rappresentazioni in lingue diverse possono avere norme diverse a causa delle differenze morfologiche e fonologiche. L'esperimento preliminare di Huang et al. con le parafrasi in inglese suggerisce che questo approccio è praticabile anche senza coppie di traduzione formalmente equivalenti, purché le viste catturino un contenuto semantico sufficientemente simile.

7.6 La struttura geometrica delle rappresentazioni indotta da LLM-JEPA

L'analisi delle proprietà geometriche dello spazio delle rappresentazioni indotte dai due regimi di addestramento fornisce un'interpretazione meccanicistica del miglioramento delle prestazioni osservato con LLM-JEPA.

La prima prospettiva è l'analisi dei valori singolari della matrice differenza Enc(Text) − Enc(Code), computata sui campioni del dataset di valutazione. I valori singolari di questa matrice misurano l'ampiezza delle direzioni lungo cui le rappresentazioni dei testi e le rappresentazioni dei codici differiscono. Se le due famiglie di embedding occupano sottospazi pressoché identici, i valori singolari saranno bassi; se occupano sottospazi ortogonali, saranno alti. I risultati mostrano che i valori singolari nel modello base e nel modello affinato con il solo obiettivo LLM sono di molti ordini di grandezza superiori a quelli del modello affinato con LLM-JEPA: le curve dei cento valori singolari dominanti per il modello base e il modello standard partono da valori dell'ordine di 10^3, mentre quelle di LLM-JEPA restano nell'ordine di 10^1. La conclusione è che la perdita JEPA vincola la trasformazione da Enc(Text) a Enc(Code) a occupare un sottospazio di dimensione effettiva

ridotta, rendendo la relazione tra le due famiglie di rappresentazioni strutturalmente più semplice.

La seconda prospettiva è l'analisi t-SNE degli embedding di Text e Code. Nel modello base, le nuvole di punti corrispondenti alle due modalità si sovrappongono in modo disordinato, senza struttura evidente. Dopo l'affinamento con il solo obiettivo LLM, la struttura si deteriora ulteriormente: la procedura di affinamento spinge i token verso ottimizzatori locali della cross-entropia che non preservano la geometria dello spazio delle rappresentazioni. Dopo l'affinamento con LLM-JEPA, emerge invece una struttura chiaramente visibile: le nuvole dei Text e dei Code si organizzano in cluster corrispondenti, con la nube del codice quasi identicamente traslata rispetto alla nube del testo lungo una direzione comune. Questa organizzazione è coerente con l'ipotesi di una trasformazione approssimativamente lineare tra le due famiglie di embedding, che viene confermata quantitativamente dal calcolo dell'errore di regressione ai minimi quadrati.

L'errore della miglior trasformazione lineare $\min_R \|\text{Enc}(\text{Text}) \cdot R - \text{Enc}(\text{Code})\|_2^2$ è pari a 4,47 per LLM-JEPA con k=1 e a 4,04 con k=0, contro 3035 per il modello standard e 3953 per il modello base. La riduzione di tre ordini di grandezza indica che, dopo l'addestramento con LLM-JEPA, l'intera relazione tra gli spazi di embedding delle due modalità è catturabile da una singola matrice lineare. Questa semplificazione strutturale è la manifestazione geometrica del miglioramento delle prestazioni: un predittore a valle non deve apprendere relazioni non lineari complesse tra testo e codice, ma può sfruttare la struttura quasi lineare indotta dall'obiettivo JEPA.

7.7 Il ruolo del predittore e la natura della competizione tra obiettivi

L'analisi dell'ablazione estesa con il parametro γ, che pesa il contributo relativo del termine generativo rispetto al termine JEPA, chiarisce la funzione specifica di ciascun componente della perdita composita. Nella formulazione generalizzata:

$$\mathcal{L}_{\text{LLM-JEPA}} = \gamma \times \sum_{\ell=2}^{L} \mathcal{L}_{\text{LLM}}(\text{Text}_{1:\ell-1}, \text{Text}_{\ell}) + \lambda \times d(\text{Pred}(\text{Enc}(\text{Text})), \text{Enc}(\text{Code}))$$

il rapporto γ/λ viene variato mantenendo $\max(\gamma, \lambda) = 1$ per conservare un tasso di apprendimento equivalente tra le configurazioni. Quando $\gamma = 0$ (nessun contributo del termine generativo), il modello affinato produce esclusivamente output vuoti: la capacità generativa collassa completamente. Questo risultato dimostra che il termine JEPA, da solo, non è sufficiente a mantenere le capacità linguistiche del modello: esso non porta gradiente nelle direzioni necessarie a far sì che il modello produca token coerenti con l'ingresso. Il termine generativo $\mathcal{L}_{\text{LLM}}$ è pertanto indispensabile, e l'obiettivo JEPA agisce come regolarizzatore dell'obiettivo primario, non come sostituto.

Con $\gamma = 0{,}1$ e $\lambda = 1{,}0$, l'accuratezza risale a 45,80% ± 5,04%, ben al di sotto del valore ottimale di 70,42% ± 2,36% ottenuto con $\gamma = \lambda = 1{,}0$. Il quadro è coerente con un'interpretazione in cui il termine generativo funge da fondamento delle capacità linguistiche del modello, mentre il termine JEPA fornisce un segnale supplementare che

affina la struttura dello spazio latente nelle direzioni rilevanti per la comprensione della dualità semantica tra testo e codice. Il predittore Pred(·) — implementato attraverso token speciali preposti all'ingresso e con pesi condivisi con il modello principale — apprende a proiettare la rappresentazione del testo verso la regione dello spazio latente occupata dalla rappresentazione del codice corrispondente, costruendo così un ponte geometrico che il predittore a valle può attraversare agevolmente.

Capitolo 8 — Architettura della loss composita e gestione dell'overhead computazionale

8.1 Il codificatore: la scelta dello stato nascosto terminale

La prima decisione implementativa che definisce l'architettura di LLM-JEPA riguarda come estrarre una rappresentazione vettoriale dell'intera sequenza di ingresso a partire dalle attivazioni interne del modello linguistico. La soluzione adottata è tecnicamente minimalista ma concettualmente motivata: si utilizza lo stato nascosto del token terminale all'uscita dell'ultimo strato del modello come embedding dell'intera sequenza. Formalmente, dato un ingresso testuale Text = (t_1, t_2, …, t_L), l'embedding contestuale è:

Enc(Text) = h_L^{(last)},

dove h_L^{(last)} ∈ ℝ^{D_model} è il vettore di attivazione all'ultimo strato per il token t_L.

Questa scelta è allineata con la pratica consolidata del "probing" dei LLM, in cui la rappresentazione dell'ultimo token è considerata quella che ha integrato il massimo di contesto attraverso i meccanismi di auto-attenzione causale. In un transformer autoregressivo, il token terminale è l'unico che ha potenzialmente accesso all'informazione contenuta in tutti i token precedenti attraverso la finestra di attenzione; la sua rappresentazione all'ultimo strato sintetizza pertanto la sequenza nella direzione in cui il modello ha imparato a integrare l'informazione per la predizione del token successivo.

Un aspetto tecnico rilevante è che questa scelta non richiede alcuna modifica all'architettura del modello: non si aggiungono strati di pooling, token CLS o meccanismi di aggregazione speciali. Il codificatore coincide con il forward pass standard del LLM, e l'embedding ne è un sottoprodotto immediato. Questa proprietà di trasparenza architetturale semplifica enormemente l'integrazione di LLM-JEPA con modelli preaddestrati esistenti: qualunque LLM basato su transformer con attenzione causale è già, implicitamente, un codificatore JEPA compatibile con questa costruzione.

Le due sequenze — Text e Code — vengono elaborate attraverso una singola passata in avanti grazie all'uso di una maschera di attenzione personalizzata, un aspetto che verrà trattato in dettaglio nella sezione successiva. Dal punto di vista del codificatore, le due sequenze sono indipendenti: Enc(Text) non vede alcun token di Code e viceversa, poiché la

struttura causale a blocchi della maschera garantisce che l'embedding finale di ciascuna sequenza rispecchi esclusivamente il proprio contenuto.

8.2 La metrica di confronto: la superiorità della similarità coseno

La scelta della metrica $d(\cdot,\cdot)$ con cui confrontare l'embedding predetto Pred(Enc(Text)) con l'embedding di riferimento Enc(Code) non è banale: metriche diverse inducono geometrie diverse sullo spazio delle rappresentazioni e producono segnali di addestramento qualitativamente distinti.

La similarità coseno è definita come:

$$d_{\cos}(u, v) = 1 - u^\top v / (\|u\|_2 \cdot \|v\|_2)$$

e misura la distanza angolare tra i due vettori, indipendentemente dalla loro norma. La scelta della similarità coseno invece della norma ℓ_2 o dell'errore quadratico medio (MSE) è motivata dalla pratica consolidata nel dominio visivo, dove la letteratura JEPA ha mostrato la superiorità della metrica angolare rispetto alle metriche euclidee per la costruzione di spazi di embedding semanticamente coerenti.

I risultati dell'ablazione confermano questa superiorità anche nel dominio linguistico. Con la norma ℓ_2 come metrica, l'accuratezza crolla a 2,22% ± 0,07%: la minimizzazione di $\|\text{Pred}(\text{Enc}(\text{Text})) - \text{Enc}(\text{Code})\|_2$ incentiva il modello a collassare entrambi gli embedding verso il vettore nullo, dove la distanza euclidea è trivialmente zero. Questo collasso rappresentazionale rispecchia un problema ben noto nelle metriche euclidee applicata a reti neurali: senza una normalizzazione esplicita delle norme, la funzione obiettivo favorisce soluzioni degeneri. La similarità coseno, operando sulla direzione dei vettori e non sulla loro ampiezza, è immune da questo collasso: la sua normalizzazione implicita forza il modello a produrre embedding con direzioni informative.

Con l'errore quadratico medio (MSE), l'accuratezza raggiunge 70,64% ± 2,05%, valore comparabile alla similarità coseno (71,46% ± 1,34%) ma con varianza leggermente superiore. La perdita InfoNCE, introdotta come alternativa contrastiva con temperatura $\tau = 0{,}07$, produce risultati particolarmente deludenti: 34,40% ± 6,10%, con deviazione standard molto elevata. La perdita InfoNCE tratta tutti gli altri campioni del minibatch come negativi, creando una pressione verso la separazione dei diversi campioni nello spazio latente; questa pressione, in assenza di un meccanismo specifico per allineare i positivi, può compromettere la struttura locale dell'embedding che è invece essenziale per la qualità delle rappresentazioni.

8.3 Il predittore a pesi condivisi: la topologia del token [PRED]

Il predittore Pred(·) è implementato attraverso un meccanismo che sfrutta la struttura autoregressiva del LLM stesso, evitando l'aggiunta di un modulo separato con parametri propri. L'idea è di appendere $k \in \{0, 1, \ldots, K\}$ token speciali [PRED] alla fine della

sequenza Text, e di utilizzare l'embedding del token finale [PRED] all'ultimo strato come output del predittore:

Pred(Enc(Text)) = h_{L+k}^{(last)},

dove L è la lunghezza della sequenza Text e h_{L+k}^{(last)} è l'attivazione dell'ultimo token [PRED] all'ultimo strato del transformer.

Questa costruzione ha una proprietà preziosa: i pesi del predittore sono identici a quelli del modello principale, poiché il token [PRED] elabora l'informazione attraverso gli stessi strati di attenzione e i medesimi parametri feed-forward. Non vi è alcun overhead in termini di parametri aggiuntivi, e la capacità espressiva del predittore deriva interamente dalla profondità del transformer e dal numero k di token [PRED] preposti. Con k = 0, il predittore è triviale: Pred(x) = x, e l'obiettivo diventa semplicemente la minimizzazione della distanza coseno tra Enc(Text) e Enc(Code) senza alcuna trasformazione intermedia. Con k > 0, il meccanismo di auto-attenzione causale consente ai token [PRED] di elaborare ulteriormente l'embedding dell'ingresso, apprendendo una trasformazione non lineare dello spazio latente che può essere regolata attraverso la scelta di k.

La griglia di ricerca degli iperparametri (k, λ) ∈ {0,1,2,3,4} × {0,5,1,2,4} mostra che il massimo delle prestazioni non cade sistematicamente né per k alti né per k bassi: la mappa delle accuratezze sulla griglia è irregolare, con massimi locali in posizioni diverse a seconda del modello e del dataset. Tuttavia, il pattern aggregato suggerisce che celle adiacenti nella griglia producono accuratezze simili, il che rende plausibile una strategia di ricerca per vicinato piuttosto che una ricerca esaustiva completa.

La scelta di preporre i token [PRED] alla fine della sequenza — piuttosto che all'inizio — è motivata dalla causalità dell'attenzione: un token [PRED] preposto all'inizio avrebbe accesso solo all'informazione dei token precedenti (nessuno, nel caso del primo token), mentre un token apposto alla fine ha accesso all'intera sequenza attraverso la finestra di attenzione causale. L'ablazione che confronta le due disposizioni mostra che la disposizione in coda è superiore (71,46% contro 68,07% con il token preposto all'inizio), confermando l'importanza dell'accesso contestuale completo.

8.4 La maschera di attenzione personalizzata a blocchi causali

Il principale ostacolo implementativo nell'applicare l'obiettivo JEPA ai LLM è ottenere gli embedding di Text e Code in modo indipendente: le due sequenze non devono "vedersi" a vicenda durante la codifica, perché altrimenti la rappresentazione di Code conterrebbe informazione proveniente da Text, compromettendo l'interpretazione dell'obiettivo come predizione da una vista all'altra.

La soluzione adottata da Huang et al. consiste nel concatenare le due sequenze in un unico ingresso di lunghezza combinata e utilizzare una maschera di attenzione additiva che implementa la causalità separatamente per ciascun blocco. La maschera standard causale per una sequenza di lunghezza totale n è una matrice triangolare inferiore dove l'elemento (i,j) vale 0 se j ≤ i (il token i può attendere al token j) e $-\infty$ se j > i (il token i non può attendere al

token j). La maschera a blocchi causali estende questa struttura: si inizializza la matrice di dimensione (batch_size × 2, 1, seq_length, seq_length) a $-\infty$ uniformemente, e si inseriscono maschere triangolari inferiori separatamente per il blocco Text e per il blocco Code:

mask[i, :, t_start : t_start + t_size, t_start : t_start + t_size] = maschera_triangolare(t_size)
mask[i, :, c_start : c_start + c_size, c_start : c_start + c_size] = maschera_triangolare(c_size)

Il risultato è una maschera che consente l'auto-attenzione causale all'interno di ciascun blocco ma blocca qualunque flusso di informazione tra i due blocchi: Text non può attendere Code e Code non può attendere Text. L'embedding finale di Text e l'embedding finale di Code sono pertanto statisticamente indipendenti condizionatamente ai rispettivi contenuti, che è esattamente la proprietà richiesta dall'obiettivo JEPA per garantire che la predizione sia non banale.

Questa implementazione consente di ottenere entrambi gli embedding in due passate in avanti invece di tre: nella prima passata si elabora la sequenza concatenata con la maschera a blocchi, ottenendo simultaneamente Enc(Text) e Enc(Code); nella seconda passata si elabora la sequenza Text con i token [PRED] appesi per ottenere Pred(Enc(Text)). Senza la maschera a blocchi, sarebbero necessarie tre passate distinte (Text, Code, Text+[PRED]), con un overhead del 200% rispetto all'addestramento standard. La soluzione con maschera a blocchi riduce questo overhead al 100%: ogni epoca di addestramento con LLM-JEPA costa il doppio di una con il solo obiettivo LLM.

8.5 Il loss dropout come meccanismo di riduzione dell'overhead

Per rendere LLM-JEPA praticabile in scenari computazionalmente vincolati, Huang et al. introducono il loss dropout (LD), un meccanismo di campionamento stocastico del termine JEPA durante l'addestramento. Con tasso di dropout LD = α, ogni minibatch applica il termine JEPA con probabilità $1-\alpha$ e lo scarta con probabilità α; nei minibatch scartati, si esegue solo il forward pass standard per il termine generativo, risparmiando l'intera passata aggiuntiva necessaria per calcolare Enc(Code) e Pred(Enc(Text)).

L'analisi del costo computazionale mostra che con LD = α il costo per epoca è $(2 - \alpha)$ volte quello dell'addestramento standard, in quanto una frazione α dei minibatch costa come il normale addestramento LLM (1 passata) e la rimanente frazione $1-\alpha$ costa il doppio (2 passate). Il costo totale è dunque:

costo_per_epoca = $\alpha \times 1 + (1-\alpha) \times 2 = 2 - \alpha$

moltiplicato per il costo dell'addestramento standard. Con LD = 0 (nessun dropout), il costo è 2×; con LD = 0,5, è 1,5×; con LD = 0,75, è 1,25×.

I risultati empirici sulle curve accuratezza-PFLOPs mostrano che LLM-JEPA converge più rapidamente del basale standard a parità di PFLOPs totali: già nelle prime epoche, con lo stesso budget computazionale, LLM-JEPA raggiunge accuratezze superiori. Questo vantaggio di efficienza si accentua ulteriormente con il loss dropout: le curve LD=0,5 e

LD=0,75 dominano la curva LD=0 nello spazio accuratezza-PFLOPs per la maggior parte del training. La ragione è che il loss dropout, aumentando la frequenza delle iterazioni con il solo obiettivo generativo, mantiene il modello più agevolmente nella traiettoria di discesa del gradiente dell'obiettivo LLM, mentre le iterazioni con il termine JEPA forniscono segnali di strutturazione intermittenti che sono comunque sufficienti a indurre l'organizzazione geometrica desiderata.

L'osservazione empirica che scalare λ proporzionalmente a $1/(1-\alpha)$ — ovvero mantenere il prodotto $\lambda(1-\alpha)$ approssimativamente costante — fornisce una regola euristica utile per il coordinamento congiunto di λ e LD. Questa invarianza approssimativa ha un'interpretazione intuitiva: il segnale JEPA medio per unità di costo computazionale è proporzionale a $\lambda(1-\alpha)$, e il comportamento del modello è governato principalmente da questa quantità piuttosto che dai due iperparametri separatamente. La regola non è un teorema formale, ma si rivela sufficientemente robusta da guidare il tuning pratico senza richiedere una ricerca sulla griglia completa (k, λ, LD).

8.6 Il protocollo sperimentale e la significatività statistica

La solidità dei risultati presentati da Huang et al. è garantita da un protocollo sperimentale rigoroso che controlla le principali sorgenti di variabilità. Ogni configurazione sperimentale viene ripetuta cinque volte con seed distinti appartenenti all'insieme {82, 23, 37, 84, 4}, e la significatività statistica delle differenze tra LLM-JEPA e il basale è valutata attraverso un test t appaiato a una coda (paired, one-tailed t-test). La scelta del test appaiato è appropriata: i modelli baseline e LLM-JEPA condividono lo stesso seed per ogni esecuzione, il che riduce la variabilità inter-campione e aumenta la potenza del test.

La procedura di selezione degli iperparametri segue uno schema a due stadi. Nel primo stadio si identifica il tasso di apprendimento ottimale $lr \in \{10^{-5}, 2\cdot10^{-5}, 4\cdot10^{-5}, 8\cdot10^{-5}\}$ per il basale standard, selezionando il valore che massimizza l'accuratezza del basale dopo quattro epoche sulla coppia (modello, dataset) data. Questo stadio garantisce che il confronto avvenga tra le migliori versioni di ciascun approccio, e non penalizzi il basale per un tasso di apprendimento sub-ottimale. Nel secondo stadio, mantenendo fisso il tasso di apprendimento ottimale del basale, si ricerca sulla griglia $(k, \lambda) \in \{0,1,2,3,4\} \times \{0,5,1,2,4\}$ la configurazione ottimale per LLM-JEPA. In tal modo si garantisce che la differenza di prestazioni non sia attribuibile a una selezione di iperparametri più favorevole per il metodo proposto.

Le metriche di accuratezza variano in funzione del task: per NL-RX-SYNTH e NL-RX-TURK si utilizza la corrispondenza esatta tra l'espressione regolare generata e quella di riferimento; per GSM8K, la corrispondenza esatta del risultato numerico finale; per Spider, la corrispondenza esatta del risultato dell'esecuzione della query generata sul database. Queste metriche di corrispondenza esatta sono particolarmente severe: anche una piccola differenza sintattica che preserva la semantica viene penalizzata allo stesso modo di un errore semantico grave. La scelta di metriche così conservative rende il vantaggio di LLM-JEPA ancora più significativo, poiché i guadagni osservati non possono essere attribuiti a criteri di valutazione permissivi.

8.7 Il fine-tuning con LoRA e la generalizzazione al regime a bassa capacità

Un'ulteriore prospettiva di validazione è fornita dagli esperimenti con l'affinamento LoRA (Low-Rank Adaptation), tecnica che riduce il numero di parametri addestrabilità mediante decomposizione a basso rango delle matrici di aggiornamento dei pesi. L'analisi a diversi rank LoRA — 32, 64, 128, 256, 512 — mostra che il vantaggio di LLM-JEPA si mantiene sistematico a tutti i rank esaminati e si accentua nei regimi di bassa capacità.

Con rank 32, il basale raggiunge 6,09% ± 0,55% mentre LLM-JEPA ottiene 7,45% ± 1,87%; con rank 512, il basale è a 50,18% ± 5,15% e LLM-JEPA a 72,41% ± 2,94%. Crucialmente, con rank 512 (corrispondente al 22,59% dei parametri addestrabilità), LLM-JEPA raggiunge un'accuratezza paragonabile all'affinamento completo con LLM-JEPA (70,42%), mentre il basale LoRA con rank 512 rimane significativamente al di sotto dell'affinamento completo con il solo obiettivo LLM (50,18% contro 57,29%). Questo risultato suggerisce che l'obiettivo JEPA svolge una funzione di regolarizzazione geometrica particolarmente efficace nel regime di bassa capacità: la pressione verso una struttura quasi lineare dell'embedding space compensa in parte la limitata espressività del modello LoRA, guidando l'ottimizzazione verso soluzioni più generalizzabili.

L'analisi della resistenza all'overfitting nel regime LoRA completa il quadro: mentre il basale standard mostra chiari segnali di overfitting nelle epoche tardive dell'affinamento, con picchi di accuratezza seguiti da declini, LLM-JEPA tende a migliorare monotonicamente o a stabilizzarsi su un plateau. Questa resistenza è particolarmente pronunciata con la configurazione LoRA rispetto all'affinamento completo, probabilmente perché la limitata capacità del modello LoRA lo rende più vulnerabile all'overfitting e l'obiettivo JEPA fornisce in questo contesto un segnale di regolarizzazione proporzionalmente più influente.

Capitolo 9 — Loss Dropout e Gestione dell'Overhead Computazionale in LLM-JEPA

9.1 Il nodo del costo computazionale come ostacolo strutturale

Ogni proposta di modifica al regime di addestramento di un modello linguistico di grandi dimensioni deve fare i conti, prima ancora che con le questioni di natura teorica, con il vincolo pratico delle risorse di calcolo. L'architettura LLM-JEPA, nella sua formulazione canonica esposta nel capitolo precedente, introduce un overhead non trascurabile rispetto all'addestramento standard basato sulla sola previsione del token successivo. Comprendere l'origine esatta di questo sovraccarico, quantificarlo con precisione e ragionare sulle strategie per attenuarlo costituisce pertanto un passaggio obbligato se si intende valutare la portata reale della proposta e la sua scalabilità verso regimi di pre-addestramento su corpora di dimensioni industriali.

La sorgente primaria dell'aggravio computazionale risiede nella necessità di ottenere, per ogni campione di addestramento, la rappresentazione latente di entrambe le viste — il testo in linguaggio naturale e il codice corrispondente, o più in generale le due proiezioni complementari della medesima conoscenza sottostante. Come illustrato nel capitolo precedente, la soluzione adottata da Huang, LeCun e Balestriero ricorre a una maschera di attenzione personalizzata che separa causalmente i blocchi di token all'interno di una singola finestra contestuale, consentendo di ottenere le due rappresentazioni Enc(Text) ed Enc(Code) in due passaggi in avanti anziché in tre. Ciononostante, il costo per iterazione rimane pari al doppio di quello richiesto dall'addestramento con la sola perdita generativa $\mathcal{L}$_LLM, poiché ogni mini-batch richiede un passaggio in avanti aggiuntivo rispetto al caso base.

Per essere precisi, si consideri il conteggio delle operazioni floating-point per epoca. Indicando con C_base il costo computazionale di un'epoca di addestramento standard — calcolato come somma dei costi di tutti i passaggi in avanti e all'indietro su tutti i mini-batch — si ha che il costo di un'epoca con LLM-JEPA nella sua formulazione integrale ammonta a $2 \cdot C_{base}$. Il fattore moltiplicativo due riflette esattamente il raddoppio dei passaggi in avanti necessari per il calcolo del termine di predizione JEPA nello spazio degli embedding, indipendentemente dal valore dell'iperparametro λ che bilancia i due contributi alla perdita composita.

Questo raddoppio non è banale. Per modelli dell'ordine del miliardo di parametri, come quelli considerati negli esperimenti descritti nel paper — Llama-3.2-1B-Instruct, Gemma-2-2b-it, OpenELM-1_1B-Instruct, OLMo-2-0425-1B-Instruct — il costo di un singolo passaggio in avanti sull'intera sequenza concatenata è già sostanziale. Raddoppiarlo significa dimezzare il numero di epoche o di esempi che possono essere processati a parità di budget energetico e temporale, oppure accettare tempi di addestramento doppi per conservare invariato il numero di aggiornamenti dei pesi.

9.2 La tecnica del loss dropout: formulazione e interpretazione

Per affrontare questa tensione tra qualità delle rappresentazioni apprese e costo computazionale, Huang, LeCun e Balestriero introducono quello che denominano *random JEPA-loss dropout* (LD), una tecnica di regolarizzazione stocastica applicata non ai parametri del modello bensì alla funzione obiettivo stessa. L'idea consiste nell'omettere il termine JEPA dalla perdita composita con probabilità α durante l'addestramento, dove $\alpha \in [0, 1)$ è un iperparametro denominato tasso di dropout della perdita.

Formalmente, sia LD = α il tasso di dropout scelto. A livello di mini-batch, ad ogni iterazione si estrae un valore booleano $b \sim \text{Bernoulli}(1 - \alpha)$: se $b = 1$, la perdita calcolata è la perdita composita completa

$$\mathcal{L}_{\text{LLM-JEPA}} = \Sigma_{\ell=2}^{L} \mathcal{L}_{\text{LLM}}(\text{Text}_{1:\ell-1}, \text{Text}_{\ell}) + \lambda \cdot d(\text{Pred}(\text{Enc}(\text{Text})), \text{Enc}(\text{Code}))$$

mentre se $b = 0$ si utilizza esclusivamente il termine generativo

$\mathcal{L}_{\text{LLM-JEPA}} \mid \{b=0\} = \Sigma\{\ell=2\}^{L} \mathcal{L}_{\text{LLM}}(\text{Text}_{1:\ell-1}, \text{Text}_{\ell})$

La conseguenza immediata è che, nei mini-batch per cui b = 0, il calcolo aggiuntivo richiesto per Enc(Code) e per il predictor Pred(·) può essere omesso del tutto, eliminando l'overhead computazionale in quella frazione α dei passi. Il costo medio per epoca diventa quindi:

$C_{LD=\alpha} = (2 - \alpha) \cdot C_{base}$

dato che in una frazione (1 - α) dei mini-batch si effettuano due passaggi in avanti, e nella restante frazione α ne basta uno solo. Per α = 0 si recupera il caso integrale $C_{LD=0} = 2 \cdot C_{base}$, mentre per $\alpha \to 1$ il costo tende asintoticamente a C_base, riducendo LLM-JEPA al semplice addestramento generativo.

Questa formulazione ha una notevole eleganza: non richiede modifiche architetturali, non introduce parametri aggiuntivi nel modello, e può essere implementata con poche righe di codice che mascherano il termine JEPA alla frequenza desiderata. La variabilità stocastica introdotta può inoltre essere interpretata come una forma di rumore nel segnale di supervisione, analogo per certi versi al dropout applicato alle attivazioni, che in taluni contesti si è rivelato benefico per la generalizzazione.

9.3 La regola empirica di co-aggiustamento di λ e α

La soppressione stocastica del termine JEPA riduce, in valore atteso, il contributo effettivo di quel termine alla perdita aggregata lungo l'epoca. Se originariamente il termine JEPA contribuisce con peso λ per ogni mini-batch, con dropout al tasso α contribuisce con peso λ · (1 - α) in media su tutti i mini-batch. Questo suggerisce che, per mantenere costante l'influenza effettiva del termine di predizione sull'ottimizzazione, si dovrebbe compensare la riduzione di frequenza con un aumento corrispondente dell'intensità, incrementando λ proporzionalmente a 1/(1 - α).

Questa osservazione corrisponde alla regola empirica identificata dagli autori, espressa come raccomandazione operativa: mantenere il prodotto λ · (1 - α) approssimativamente costante durante la ricerca degli iperparametri. In termini simbolici:

$\lambda_{new} \cdot (1 - \alpha) \approx \lambda_{base}$

dove λ_base è il valore ottimale identificato per il caso senza dropout (α = 0). Sostituendo, si ottiene la prescrizione:

$\lambda_{new} \approx \lambda_{base} / (1 - \alpha)$

Per α = 0.5 si ha $\lambda_{new} \approx 2 \cdot \lambda_{base}$, per α = 0.75 si ha $\lambda_{new} \approx 4 \cdot \lambda_{base}$. Questa relazione ha un chiaro fondamento nell'analisi dell'ottimizzazione stocastica: il gradiente atteso rispetto ai parametri del predictor e dell'encoder è proporzionale a λ · (1 - α), e pertanto mantenere costante questo prodotto assicura che la dinamica dell'addestramento rimanga qualitativamente simile a quella del caso integrale, pur con un costo computazionale ridotto.

Gli autori precisano tuttavia che la regola costituisce una guida approssimata piuttosto che una legge esatta. Il comportamento reale dipende dalla curvatura locale della superficie di perdita, dall'interazione con il tasso di apprendimento e dalla dimensione del mini-batch, tutti fattori che possono modulare l'effetto in misura difficilmente prevedibile a priori. La regola $\lambda \cdot (1 - \alpha) \approx \text{const}$ va pertanto intesa come punto di partenza per la ricerca degli iperparametri, non come sostituto di essa.

9.4 Analisi sperimentale del loss dropout: risultati quantitativi

La validazione empirica della tecnica proposta si fonda su esperimenti condotti con il modello Llama-3.2-1B-Instruct sul dataset NL-RX-SYNTH, misurando l'accuratezza in funzione dei PetaFLOP (PFLOP) consumati complessivamente, in modo da confrontare le diverse configurazioni a parità di budget computazionale. I risultati riportati nel paper permettono di ricostruire con precisione il comportamento della tecnica.

Nella configurazione di riferimento senza dropout (LD = 0, $\lambda = 1$, k = 1), LLM-JEPA raggiunge un'accuratezza del 63.57% avendo consumato circa 14.49 PFLOP, contro il 55.60% della sola fine-tuning generativa allo stesso costo. Introducendo un tasso di dropout LD = 0.5 con lo stesso valore di $\lambda = 1$, la configurazione consuma in media $1.5 \cdot C_base$ per epoca, e a parità di PFLOP consumati riesce a processare un numero maggiore di epoche. I risultati mostrano che LD = 0.5 con $\lambda = 1$ raggiunge 64.08% a 14.49 PFLOP, superando marginalmente la variante senza dropout allo stesso costo, e 69.87% a 19.32 PFLOP.

La variante LD = 0.5 con $\lambda = 2$ — che rispetta la regola di co-aggiustamento avendo raddoppiato λ all'aumentare del dropout — produce risultati ancora più convincenti: 63.79% a 14.49 PFLOP e 70.11% a 19.32 PFLOP, sostanzialmente equivalenti o superiori a LD = 0. Analogamente, la configurazione con dropout aggressivo LD = 0.75 e $\lambda = 4$ (che rispetta la regola: $1 \cdot (1\text{-}0) = 1$, $4 \cdot (1\text{-}0.75) = 1$) raggiunge 65.32% a 14.49 PFLOP e 72.42% a 28.98 PFLOP, il valore più elevato tra tutte le configurazioni considerate all'ultimo checkpoint.

Questi dati vanno letti congiuntamente per comprenderne il significato. Il dropout elevato (LD = 0.75) consente di processare più epoche allo stesso costo, sfruttando il fatto che la convergenza di LLM-JEPA è in generale monotonicamente crescente con le epoche — almeno nel regime considerato, in cui non si osserva overfitting per la variante JEPA a differenza del solo addestramento generativo. Il vantaggio del dropout diventa così duplice: riduzione del costo per epoca e resistenza all'overfitting preservata, che consente di spingere l'addestramento oltre senza penalizzazione.

9.5 Il profilo di convergenza e la superiore efficienza campionaria

Un elemento particolarmente significativo che emerge dall'analisi dei risultati è che LLM-JEPA converge più rapidamente del baseline generativo a parità di PFLOP consumati. In termini grafici — quelli mostrati nella Figura 5 del paper — la curva di accuratezza in funzione dei PFLOP di LLM-JEPA si trova sistematicamente al di sopra di quella

dell'addestramento standard per tutti i valori dell'asse orizzontale considerati, indipendentemente dalla configurazione del dropout.

Questo comportamento suggerisce che il termine JEPA non si limita a fornire un segnale di supervisione aggiuntivo che si accumula nel tempo, ma organizza lo spazio delle rappresentazioni in modo tale da rendere l'ottimizzazione del termine generativo più efficiente. I risultati della Figura 3 del paper — che mostra come la perdita sul token successivo (loss_LLM) sia sostanzialmente identica nelle due varianti di addestramento, con e senza JEPA — rendono questa interpretazione più precisa: il termine JEPA aggiunge struttura allo spazio latente senza interferire con il segnale generativo, e tale struttura aggiuntiva si traduce in una migliore capacità di generalizzazione pur in assenza di riduzioni della perdita sul training set.

La combinazione di queste proprietà — convergenza rapida, resistenza all'overfitting, strutturazione latente — rende il loss dropout una soluzione particolarmente attraente per il regime di pre-addestramento su corpora di grandi dimensioni, dove il numero di passi di ottimizzazione è tipicamente molto elevato e il costo computazionale è il vincolo dominante. Gli autori indicano esplicitamente questo come una delle direzioni future di sviluppo della ricerca, ritenendo che i risultati preliminari sul pre-addestramento — discussi altrove nel paper — siano sufficientemente incoraggianti da giustificare un'estensione a scala piena.

18.11 Roadmap implementativa per LLM-Var-JEPA: componenti critici e scelte di design

L'obiettivo unificato LLM-Var-JEPA proposto nel Capitolo 18 richiede, per la sua realizzazione pratica, soluzioni a sfide implementative concrete. Il context encoder variazionale q_φ(s_text | Text) può essere implementato aggiungendo due teste di proiezione lineari (media μ e log-varianza log σ^2) sopra l'ultimo layer di un LLM preaddestrato, con reparametrization trick $s_text = \mu + \sigma \odot \varepsilon$ per la differenziabilità. Il costo computazionale aggiuntivo è $O(D^2_model)$ per le proiezioni — trascurabile rispetto al transformer principale.

La gestione della stabilità di addestramento richiede gradient clipping separato per i tre componenti della perdita ($\mathcal{L}$_LLM, $\mathcal{L}$^KL_{s_text}, $\mathcal{L}$_JEPA) e KL annealing per il termine variazionale. Il metodo PCGrad (Yu et al. 2020) — che proietta i gradienti in conflitto su sottospazi ortogonali — potrebbe eliminare l'interferenza distruttiva tra i tre segnali. La costruzione di viste appropriate per il preaddestramento su corpus generici rimane la sfida più critica: l'uso di modelli di paraphrase generation per creare viste sintetiche è praticabile ma introduce dipendenza dalla qualità del modello di parafrasi, creando un potenziale bootstrapping migliorabile iterativamente.

18.12 Leggi di scala per LLM-Var-JEPA: previsioni teoriche e verifiche sperimentali

Le scaling laws di Kaplan et al. (2020) descrivono come le prestazioni LLM scalano con parametri N, dati D e budget C. Un'analisi preliminare suggerisce che LLM-JEPA riduce

il coefficiente C_∞ (piso di prestazione irraggiungibile) producendo un offset positivo rispetto al baseline per qualsiasi scala. Questa ipotesi è coerente con l'interpretazione di LLM-JEPA come regolarizzazione strutturale: la struttura geometrica dello spazio latente è una proprietà qualitativa che riduce la quantità di dati necessaria per una data prestazione, senza necessariamente cambiare l'esponente delle scaling laws.

I dati di Huang et al. su modelli 1B, 3B, 7B e 8B mostrano un miglioramento relativo nell'ordine del 10-30% per tutti i modelli. Una verifica sistematica richiederebbe esperimenti su almeno 5-6 punti di scala tra 100M e 100B parametri con dataset standardizzati. La regola empirica $\lambda^* \propto N_params^{\alpha}$ — dove modelli più grandi richiedono un segnale JEPA più intenso per ristrutturare lo spazio latente già consolidato — potrebbe essere formalizzata in una scaling law estesa che incorpora λ come parametro aggiuntivo accanto a N, D e C.

9.6 La maschera di attenzione personalizzata come infrastruttura abilitante

Per comprendere pienamente la tecnica del loss dropout e il modo in cui riduce l'overhead a due passaggi in avanti anziché a tre, è necessario tornare al meccanismo della maschera di attenzione personalizzata. Come descritto nel capitolo precedente, la realizzazione di LLM-JEPA senza dropout richiede di ottenere separatamente Enc(Text) ed Enc(Code), dove entrambe le codifiche devono essere calcolate senza che i token di una sequenza attendano a quelli dell'altra — violazione che si verificherebbe con la semplice concatenazione e la maschera causale standard.

La soluzione adottata è una maschera causale a blocchi: all'interno di ogni blocco (Text o Code), la causalità è preservata nel senso che il token in posizione i può attendere a tutti i token $j < i$ dello stesso blocco; tra i blocchi, invece, nessun token di un blocco può attendere a token dell'altro. Formalmente, per un batch di dimensione batch_size con due sequenze di lunghezza t_size e c_size che iniziano rispettivamente alle posizioni t_start e c_start nella finestra contestuale concatenata, la maschera additiva è definita come una matrice $M \in \mathbb{R}^{(t_size + c_size) \times (t_size + c_size)}$ con:

$M[i, j] = 0$ se $(i, j) \in$ blocco_Text $\times$ blocco_Text e $i \geq j$ $M[i, j] = 0$ se $(i, j) \in$ blocco_Code $\times$ blocco_Code e $i \geq j$
$M[i, j] = -\infty$ altrimenti

L'embedding finale dell'ultimo token della sequenza Text costituisce Enc(Text), e analogamente per Enc(Code). Con questa costruzione, entrambe le codifiche sono ottenibili in un unico passaggio in avanti — il che riduce il totale a due passaggi (uno per il calcolo degli embedding, uno per la perdita generativa con il modello autoregressivo) rispetto ai tre che sarebbero necessari con passaggi separati. Quando il dropout è attivo per un dato mini-batch ($b = 0$), il secondo passaggio in avanti per il calcolo degli embedding può essere omesso del tutto, portando il costo di quell'iterazione a un solo passaggio, identico al caso dell'addestramento generativo standard.

Questo chiarisce perché il fattore di costo per epoca con dropout al tasso α è esattamente $(2 - \alpha)$: in proporzione $(1 - \alpha)$ dei mini-batch si effettuano due passaggi, e in proporzione α ne basta uno solo, da cui il costo medio per mini-batch è $(1 - \alpha) \cdot 2 + \alpha \cdot 1 = 2 - \alpha$, proporzionale a C_base a parametri architetturali fissati.

9.7 Verso il pre-addestramento a scala piena: prospettive e limitazioni

La questione della scalabilità del loss dropout verso regimi di pre-addestramento su corpora di miliardi di token rimane aperta al termine degli esperimenti descritti nel paper. Gli autori riconoscono che i risultati sul pre-addestramento, pur incoraggianti, sono stati ottenuti su dataset di dimensioni moderate — il dataset NL-RX-SYNTH per gli esperimenti principali, e il dataset cestwc/paraphrase per l'esperimento sul pre-addestramento con parafrasi — e che estendere le conclusioni a corpora di scala web richiederebbe esperimenti dedicati.

Restano in particolare due questioni non pienamente risolte. La prima concerne la sensibilità degli iperparametri k e λ al variare della scala: la Figura 7 del paper mostra che la superficie di accuratezza in funzione di (k, λ) può presentare comportamenti complessi, con picchi e valli in posizioni non prevedibili a priori, il che rende il tuning di questi parametri costoso in termini di risorse. La regola di co-aggiustamento $\lambda \cdot (1 - \alpha) \approx \text{const}$ offre una guida approssimata ma non elimina la necessità di una ricerca sistematica.

La seconda questione riguarda l'assunzione di disponibilità di viste non banali. Come sottolineato dagli autori, il successo di LLM-JEPA dipende in modo critico dalla presenza di coppie di sequenze che siano genuinamente due proiezioni della stessa conoscenza sottostante — coppie (testo, codice) nel caso dei dataset considerati, parafrasi nel caso dell'esperimento aggiuntivo. Per dataset generici, privi di questa struttura, sarebbe necessario sviluppare un meccanismo analogo all'augmentation dei dati nel caso visivo, capace di generare automaticamente viste non banali a partire da testi arbitrari. Questo rimane un problema aperto e rappresenta la principale limitazione della proposta nella sua formulazione attuale.

9.8 Il loss dropout come curriculum stocastico: interpretazione nella teoria dell'apprendimento

L'interpretazione del loss dropout come semplice riduzione dell'overhead computazionale, pur corretta, non coglie la sua funzione più profonda come meccanismo di curriculum di addestramento. La nozione di curriculum learning (Bengio et al. 2009) si basa sull'osservazione che l'ordine in cui i campioni o gli obiettivi vengono presentati durante l'addestramento influenza significativamente la qualità delle rappresentazioni apprese: presentare esempi in ordine di difficoltà crescente, o alternare obiettivi semplici con obiettivi complessi, può produrre convergenza più rapida e soluzioni di qualità superiore rispetto alla presentazione casuale.

Il loss dropout implementa implicitamente un curriculum a livello di obiettivi: nei batch dove $LD = 0$ (JEPA term attivo), il modello riceve il segnale di strutturazione latente più

ricco ma computazionalmente costoso; nei batch dove LD = 1 (solo obiettivo LLM), riceve il segnale generativo standard più semplice. L'alternanza stocastica tra questi due regimi ha un effetto regolarizzante che va al di là della semplice media ponderata degli obiettivi: nei batch "LLM-only", i parametri del modello vengono aggiornati esclusivamente sulla base della perdita generativa, il che preserva le capacità linguistiche acquisite nei batch precedenti senza interferenza del gradiente JEPA. Nei batch "JEPA-active", i parametri vengono aggiornati congiuntamente per ottimizzare sia la coerenza cross-modale nello spazio latente sia la generazione linguistica. Questa separazione temporale degli obiettivi riduce il rischio di interferenza distruttiva tra i due gradienti, analogo al catastrophic forgetting ma a livello di obiettivi piuttosto che di compiti.

L'analisi del paesaggio della perdita sotto loss dropout può essere formalizzata attraverso il framework delle stochastic weight averaging (SWA, Izmailov et al. 2018). Il punto fisso dell'ottimizzazione con loss dropout non è il minimo di una funzione di perdita con peso fisso λ, ma il punto di convergenza dell'iterazione stocastica $\theta(t+1) = \theta(t) - \eta(t) \cdot [b(t) \cdot \nabla\mathcal{L}_\text{LLM-JEPA} + (1 - b(t)) \cdot \nabla\mathcal{L}_\text{LLM}]$, dove $b(t) \sim \text{Bernoulli}(1 - \alpha)$. L'analisi dell'ensemble implicito di modelli visitati lungo questa traiettoria --- che è l'idea centrale di SWA --- suggerisce che il punto fisso corrisponde alla media di un ensemble di modelli: alcuni con l'obiettivo JEPA pienamente attivo, altri con solo l'obiettivo LLM. Questa media produce rappresentazioni che combinano le proprietà di strutturazione latente (dal componente JEPA) con la robustezza generativa (dal componente LLM), risultando in una qualità superiore rispetto a qualunque punto fisso a coefficiente fisso.

9.9 Scaling del loss dropout verso il preaddestramento industriale: analisi dei costi e benefici

La proiezione di LLM-JEPA verso il regime di preaddestramento su scala industriale --- corpus di trilioni di token, modelli di decine di miliardi di parametri --- richiede un'analisi attenta dei trade-off computazionali a questa scala. Il loss dropout con $\alpha = 0.9$ riduce l'overhead a solo 1.1× rispetto al preaddestramento standard, ma il beneficio assoluto in termini di qualità delle rappresentazioni potrebbe essere inferiore a quello osservato nel regime di fine-tuning su dataset piccoli.

Due fattori si muovono in direzioni opposte quando si scala da fine-tuning a preaddestramento. Il primo è il numero di aggiornamenti per epoch: nel preaddestramento su trilioni di token, il modello esegue circa 10^6-10^7 aggiornamenti per epoch, mentre nel fine-tuning su dataset piccoli sono tipicamente 10^2-10^3. Con un numero maggiore di aggiornamenti, anche un segnale JEPA molto raro ($\alpha = 0.99$, solo 1 batch su 100) può accumulare un contributo significativo alla strutturazione dello spazio latente. Il secondo fattore è la distribuzione dei dati: il preaddestramento su corpus generici non garantisce la disponibilità di coppie (testo, codice) strutturate come viste JEPA. Huang et al. propongono le parafrasi come alternativa, ma la qualità delle viste parassi è inferiore a quella delle coppie testo-codice, il che potrebbe richiedere un rapporto λ più alto per compensare.

Una valutazione quantitativa del punto di pareggio computazionale per il preaddestramento può essere condotta come segue. Sia C_base il costo di pre-addestrare un LLM standard su T token. Il preaddestramento con LLM-JEPA (loss dropout α = 0.9, LD) costa (2 - 0.9) · C_base = 1.1 · C_base. Se LLM-JEPA riduce il numero di token necessari per raggiungere una data qualità di embedding da T a T/f (dove f > 1 è il fattore di efficienza), allora il preaddestramento JEPA è conveniente se 1.1 · C_base / f < C_base, ovvero f > 1.1. I risultati sul fine-tuning di Huang et al. mostrano che LLM-JEPA raggiunge la stessa accuratezza del baseline standard con circa il 70% dei PFLOPs (dal confronto delle curve nella Figura 5), corrispondente a f ≈ 1.43 >> 1.1. Se questo fattore di efficienza si mantiene anche in regime di preaddestramento, LLM-JEPA con loss dropout aggressivo sarebbe computazionalmente conveniente anche a scala industriale.

Capitolo 10 — Risultati Empirici: NL-RX-SYNTH, GSM8K, Spider, HellaSwag e Famiglie di Modelli

10.1 Il protocollo sperimentale come garanzia di rigore statistico

Prima di entrare nel merito dei risultati quantitativi, è necessario descrivere con precisione il protocollo sperimentale adottato da Huang, LeCun e Balestriero, poiché la solidità metodologica di una campagna di misurazioni costituisce la premessa indispensabile per l'interpretazione delle cifre. Ogni esperimento è ripetuto cinque volte con semi casuali fissi — {82, 23, 37, 84, 4} — scelti prima dell'esecuzione e mantenuti invariati attraverso tutte le configurazioni. Questa scelta consente di condurre test t accoppiati unilaterali (paired one-tailed t-test) tra la variante baseline e LLM-JEPA, controllando per la variabilità dovuta all'inizializzazione e all'ordine dei dati piuttosto che per la variabilità assoluta delle due distribuzioni.

Il test t accoppiato è la scelta corretta in questo contesto: poiché le due condizioni sperimentali (baseline e LLM-JEPA) condividono gli stessi semi, le cinque coppie di misurazioni (accuratezza_baseline_seed_i, accuratezza_JEPA_seed_i) sono correlate positivamente, e la statistica del test sfrutta questa correlazione per aumentare la potenza statistica. La statistica del test è:

$t = (\bar{d}) / (s_d / \sqrt{n})$

dove $\bar{d}$ è la media delle differenze d_i = accuratezza_JEPA_i − accuratezza_baseline_i, s_d è la deviazione standard campionaria delle differenze, e n = 5 è il numero di ripetizioni. Con n − 1 = 4 gradi di libertà, il valore critico per un test unilaterale al livello di significatività del 5% è $t_{0.05, 4} \approx 2.13$, e quello per il 1% è $t_{0.01, 4} \approx 3.75$.

Per ciascuna coppia (modello, dataset), si adotta una procedura di selezione degli iperparametri in due fasi. Nella prima, si identifica il tasso di apprendimento ottimale $\eta \in \{1\times10^{-5}, 2\times10^{-5}, 4\times10^{-5}, 8\times10^{-5}\}$ per il baseline generativo, scegliendo il valore che massimizza l'accuratezza dopo 4 epoche. Nella seconda, si effettua una ricerca a griglia bidimensionale sugli iperparametri specifici di LLM-JEPA: $(k, \lambda) \in \{0, 1, 2, 3, 4\} \times \{0.5, 1, 2, 4\}$, per un totale di 20 configurazioni, utilizzando il medesimo tasso di apprendimento identificato per il baseline. Questa procedura garantisce che il confronto avvenga tra il miglior baseline ragionevole e la miglior configurazione LLM-JEPA, minimizzando il rischio che differenze di accuratezza riflettano semplicemente una scelta sub-ottimale degli iperparametri del baseline.

La metrica di accuratezza varia a seconda del dataset. Per NL-RX-SYNTH e NL-RX-TURK, l'accuratezza è la corrispondenza esatta (*exact match*) tra l'espressione regolare generata e quella di riferimento: ogni token della sequenza prodotta deve coincidere con il corrispondente token della ground truth, senza eccezioni. Per GSM8K, l'accuratezza è la corrispondenza esatta del risultato numerico finale del problema matematico. Per Spider, è la corrispondenza esatta del risultato di esecuzione della query SQL generata sul database di riferimento — una metrica più permissiva della corrispondenza stringa, poiché due query SQL sintatticamente diverse possono produrre risultati identici.

10.2 NL-RX-SYNTH e NL-RX-TURK: traduzione da linguaggio naturale a espressioni regolari

Il dataset NL-RX-SYNTH (Locascio et al., 2016) è il banco di prova principale degli esperimenti. Ogni campione è una coppia (descrizione in linguaggio naturale, espressione regolare) dove la descrizione è sintetica, generata programmaticamente, e l'espressione regolare è l'oggetto da apprendere. Il dataset NL-RX-TURK adotta la stessa struttura ma con descrizioni in linguaggio naturale scritte da annotatori umani tramite Amazon Mechanical Turk, risultando quindi più vario e idiomatico — e conseguentemente più difficile.

Per Llama-3.2-1B-Instruct su NL-RX-SYNTH, il baseline raggiunge 57.29% ± 5.32% (media ± deviazione standard su cinque semi), mentre LLM-JEPA con la configurazione ottimale ($k = 1$, $\lambda = 1$, $\eta = 2\times10^{-5}$) raggiunge 71.46% ± 1.34%, con un miglioramento assoluto di circa 14 punti percentuali. Il p-value del test t accoppiato è 1.0×10^{-3}, ampiamente al di sotto della soglia del 1%, confermando la significatività statistica. Vale la pena sottolineare che la deviazione standard di LLM-JEPA è notevolmente inferiore a quella del baseline: 1.34% contro 5.32%. Questa riduzione della varianza suggerisce che il termine JEPA esercita un effetto stabilizzante sull'ottimizzazione, rendendo il risultato finale meno dipendente dall'inizializzazione casuale.

Un aspetto cruciale che emerge dall'analisi delle perdite mostrata nella Figura 3 del paper è che le due varianti — baseline e LLM-JEPA — producono valori praticamente identici della perdita generativa $\mathcal{L}$_LLM lungo l'intero addestramento (le curve corrispondenti si sovrappongono nel grafico). La differenza di accuratezza — 14 punti percentuali — non può quindi essere attribuita a una migliore previsione del token successivo nel senso classico, ma deve originare dalla strutturazione aggiuntiva dello spazio latente operata dal termine JEPA. Questo punto è dirimente perché esclude l'interpretazione alternativa secondo cui

LLM-JEPA otterrebbe semplicemente un'ottimizzazione più aggressiva della stessa funzione obiettivo: le due varianti convergono allo stesso punto sulla superficie della perdita generativa, ma la varietà LLM-JEPA porta con sé una struttura latente aggiuntiva che le conferisce una capacità di generalizzazione superiore.

Per lo stesso modello su NL-RX-TURK, il baseline raggiunge 22.49% ± 1.91% e LLM-JEPA 30.94% ± 1.13% (p-value = 2.4×10^{-4}, k = 1, λ = 1). Il miglioramento assoluto di circa 8.5 punti su un dataset con variabilità linguistica maggiore è ancora molto significativo, e la deviazione standard rimane inferiore per LLM-JEPA rispetto al baseline, confermando il pattern di stabilizzazione osservato su NL-RX-SYNTH.

I risultati sul profilo temporale — accuratezza in funzione del numero di epoche, fino a sei epoche totali — mostrano un ulteriore elemento di interesse. Il baseline presenta un chiaro comportamento di overfitting dopo le prime epoche: l'accuratezza sul set di validazione raggiunge un picco e poi degrada, segnalando che il modello memorizza le specificità del set di addestramento a scapito della generalizzazione. LLM-JEPA, al contrario, mostra una progressione sostanzialmente monotonicamente crescente lungo tutte le sei epoche considerate, senza alcun segno di degradazione. La Figura 1 del paper illustra questo comportamento in modo particolarmente netto per la coppia Llama3/NL-RX-TURK, dove la curva baseline ha già raggiunto il suo picco alla seconda epoca mentre quella LLM-JEPA continua a migliorare fino alla sesta.

10.3 GSM8K: ragionamento matematico elementare

Il dataset GSM8K (Cobbe et al., 2021) è costituito da circa 8000 problemi matematici di livello elementare espressi in linguaggio naturale, con soluzioni che richiedono da due a otto passaggi di ragionamento aritmetico. La metrica è la corrispondenza esatta del risultato numerico finale.

Per Llama-3.2-1B-Instruct su GSM8K, il baseline raggiunge 32.36% ± 0.58% e LLM-JEPA 36.36% ± 0.20% nella migliore configurazione (k = 4, λ = 0.5, p-value = 9.6×10^{-5}). Il p-value estremamente basso — inferiore all'1 per mille — segnala un effetto statisticamente robusto nonostante la deviazione standard già contenuta del baseline. La riduzione ulteriore della varianza ottenuta con LLM-JEPA è ancora più pronunciata: da 0.58% a 0.20%, un fattore quasi tre.

La dimensione dell'effetto su GSM8K (circa 4 punti percentuali assoluti) è minore di quella osservata su NL-RX-SYNTH (circa 14 punti), il che è atteso: GSM8K non presenta la struttura divistacea naturale che caratterizza i dataset di traduzione testo-codice, e la coppia (testo del problema, soluzione numerica) costituisce una vista meno simmetrica di (testo, codice). La soluzione numerica è tipicamente molto più breve della descrizione del problema, e il contenuto semantico delle due sequenze è parzialmente sovrapposto ma non complementare nello stesso senso in cui lo sono un problema descritto in linguaggio naturale e la sua implementazione in codice SQL o in un'espressione regolare. Che LLM-JEPA produca comunque un miglioramento significativo suggerisce che il termine di predizione nello spazio degli embedding cattura relazioni semantiche utili anche in assenza di viste perfettamente complementari.

La griglia degli iperparametri per GSM8K con tasso di apprendimento $\eta = 2\times10^{-5}$ è mostrata nella Figura 7(a) del paper. La superficie di accuratezza è relativamente piatta nella regione di valori bassi di λ (0.5-1) e alti di k (3-4), con la configurazione ottimale a $\lambda = 0.5$, $k = 4$. Il pattern della griglia rivela che per questo dataset specifico è preferibile un predictor con più token di predizione (k elevato) e un peso relativamente modesto del termine JEPA (λ piccolo), suggerendo che il segnale di predizione nello spazio latente è utile ma deve essere bilanciato con cura per non dominare il segnale generativo.

10.4 Spider: comprensione del linguaggio naturale e generazione SQL

Il dataset Spider (Yu et al., 2018) è un corpus ad alto grado di complessità per il *text-to-SQL parsing* in domini multipli: il testo in input è una coppia (identificatore del database, domanda in linguaggio naturale) e il codice target è la query SQL che risponde alla domanda interrogando il database specificato. La metrica è la corrispondenza del risultato di esecuzione, che è più permissiva della corrispondenza stringa ma richiede comunque che la query generata produca le stesse righe della query di riferimento.

Per Llama-3.2-1B-Instruct su Spider, il baseline raggiunge 47.52% ± 2.44% e LLM-JEPA 50.55% ± 2.08% nella configurazione ottimale ($k = 3$, $\lambda = 1$, $\eta = 4\times10^{-5}$, p-value = 4.0×10^{-3}). Il miglioramento assoluto è di circa 3 punti percentuali — più contenuto che per NL-RX, il che può riflettere la maggiore complessità strutturale del compito (le query SQL in Spider coinvolgono join, aggregazioni e subquery che richiedono una comprensione precisa dello schema del database) nonché la natura diversa della vista Code, che in questo caso è una query SQL complessa piuttosto che un'espressione regolare relativamente compatta.

La configurazione ottimale con $k = 3$ indica che tre token di predizione sono sufficienti per modellare la relazione tra la rappresentazione del testo e quella della query, suggerendo che il mapping tra le due viste è abbastanza regolare da essere catturato da un predictor di complessità moderata. Il valore di $\lambda = 1$ indica che i due termini della perdita contribuiscono con pesi equivalenti, senza che sia necessario enfatizzare il termine JEPA rispetto al termine generativo.

La griglia degli iperparametri per Spider, mostrata nella Figura 7(b) del paper con $\eta = 1\times10^{-5}$, presenta una superficie più regolare rispetto a quella di GSM8K, con un plateau piuttosto esteso nella regione $\lambda \in \{0.5, 1\}$ e $k \in \{2, 3\}$. Questa maggiore regolarità può essere interpretata come un indicatore che, per dataset con una struttura di viste naturalmente complementare come Spider — dove la domanda in linguaggio naturale e la query SQL sono genuinamente due rappresentazioni della stessa interrogazione — la funzione obiettivo di LLM-JEPA è meno sensibile alla scelta precisa degli iperparametri entro un range ragionevole.

10.5 HellaSwag e NQ-Open: oltre la struttura testo-codice

Un risultato particolarmente significativo per la valutazione della portata di LLM-JEPA è il suo comportamento su dataset che non presentano la struttura naturale testo-codice sfruttata nei casi precedenti. HellaSwag (Zellers et al., 2019) è un benchmark per la comprensione del linguaggio in formato a scelta multipla: dato un contesto testuale, il sistema deve identificare il completamento più plausibile tra quattro alternative. NQ-Open (Lee et al., 2019) è un dataset di domande aperte a risposta breve.

Per adattare LLM-JEPA a HellaSwag, gli autori definiscono Text come il contesto e Code come il completamento corretto, formando una coppia di viste dove la relazione è di tipo contesto-continuazione piuttosto che descrizione-implementazione. Questa è una forma di vista meno simmetrica e meno complementare di quella testo-codice: il completamento non è una rappresentazione alternativa del contesto ma la sua naturale prosecuzione. Nonostante questa differenza strutturale, LLM-JEPA raggiunge 70.51% ± 1.20% contro il 69.40% ± 0.99% del baseline (p-value = 0.0136, $k = 3$, $\lambda = 1$, $\eta = 4\times10^{-5}$), con un miglioramento di circa 1.1 punti percentuali.

Per NQ-Open, Text è la domanda e Code è lo span di risposta, tipicamente molto più corto. LLM-JEPA raggiunge 21.59% ± 0.40% contro il 20.12% ± 0.41% del baseline (p-value = 2.44×10^{-3}, $k = 0$, $\lambda = 1024$, $\eta = 2\times10^{-5}$). La configurazione ottimale per questo dataset è notevolmente diversa dalle altre: il predictor banale ($k = 0$, cioè $\text{Pred}(x) = x$) con un peso JEPA enorme ($\lambda = 1024$) suggerisce che per coppie domanda-risposta molto asimmetriche la funzione obiettivo ottimale si approssima alla minimizzazione della distanza coseno diretta tra le rappresentazioni, senza elaborazione aggiuntiva da parte del predictor. Ciò è coerente con l'interpretazione secondo cui il predictor è necessario quando le due viste richiedono una trasformazione non banale per essere messe in corrispondenza, mentre per viste molto semplici — dove il mapping è quasi identitario — il predictor banale è sufficiente.

Un'osservazione importante per HellaSwag è che la superficie di accuratezza in funzione di λ mostra una progressione montonamente crescente fino a $\lambda = 1024$, senza alcun plateau né degradazione nell'intervallo esplorato. Questo comportamento insolito — descritto nella Figura 9 del paper — suggerisce che per questo dataset il termine JEPA è benefico in modo proporzionale alla sua intensità, almeno nel range considerato, e che il valore ottimale di λ potrebbe essere ancora più elevato. Gli autori segnalano questo come una peculiarità degna di ulteriore indagine.

10.6 Generalizzazione su quattro famiglie di modelli

La robustezza di LLM-JEPA è valutata non solo su più dataset ma anche su quattro famiglie di architetture distinte, tutte nella classe dei modelli con circa un miliardo di parametri. I risultati su NL-RX-SYNTH con la migliore configurazione per ciascun modello sono i seguenti.

Llama-3.2-1B-Instruct (Meta, Grattafiori et al., 2024): baseline 57.29% ± 5.32%, LLM-JEPA 71.46% ± 1.34%, miglioramento di 14.17 punti percentuali assoluti, configurazione ottimale $k = 1$, $\lambda = 1$, p-value = 1.0×10^{-3}.

Gemma-2-2b-it (Google DeepMind, Team et al., 2024): baseline 33.65% ± 3.24%, LLM-JEPA 43.12% ± 2.61%, miglioramento di 9.47 punti, configurazione ottimale $k = 4$, $\lambda = 2$, $\eta = 1\times10^{-5}$, p-value = 5.5×10^{-3}.

OpenELM-1_1B-Instruct (Apple, Mehta et al., 2024): baseline 12.07% ± 1.81%, LLM-JEPA 25.40% ± 2.40%, miglioramento di 13.33 punti, configurazione ottimale $k = 3$, $\lambda = 4$, $\eta = 8\times10^{-5}$, p-value = 5.1×10^{-4}.

OLMo-2-0425-1B-Instruct (Allen Institute, OLMo et al., 2024): baseline 87.09% ± 0.36%, LLM-JEPA 87.52% ± 0.29%, miglioramento di 0.43 punti, configurazione ottimale $k = 0$, $\lambda = 2$, $\eta = 8\times10^{-5}$, p-value = 2.5×10^{-3}.

Il caso di OLMo merita un commento separato. Il miglioramento assoluto è di meno di mezzo punto percentuale, ma la significatività statistica è confermata con un p-value di 2.5×10^{-3}, il che significa che l'effetto è reale anche se di piccola entità. La ragione del miglioramento contenuto è probabilmente il livello di accuratezza già molto elevato del baseline (87.09%), che riduce lo spazio disponibile per ulteriori guadagni. Il fenomeno noto come *effetto soffitto* — la tendenza dei modelli forti a saturare le metriche di valutazione su dataset che risultano relativamente semplici per le loro capacità — spiega perché un modello che già raggiunge oltre l'87% di corrispondenza esatta su espressioni regolari abbia meno margine di miglioramento rispetto a uno che parte dal 12% o dal 33%.

La configurazione ottimale di OLMo con $k = 0$ — predictor banale — è anch'essa interpretabile alla luce dell'elevata accuratezza del baseline: per un modello che già rappresenta bene le relazioni testo-codice, il termine JEPA ottimale è quello che impone semplicemente la prossimità delle rappresentazioni nello spazio latente senza richiedere una trasformazione elaborata.

10.7 Risultati per dimensioni del modello crescenti

Gli autori estendono la valutazione a modelli di dimensioni maggiori, specificatamente Llama-3.2-3B-Instruct (3 miliardi di parametri), Llama-3.1-8B-Instruct (8 miliardi di parametri) e OLMo-2-1124-7B-Instruct (7 miliardi di parametri), sempre su NL-RX-SYNTH.

Per Llama-3.2-3B-Instruct, il baseline raggiunge 74.55% ± 3.58% e LLM-JEPA 77.16% ± 3.66% ($k = 0$, $\lambda = 2$, $\eta = 2\times10^{-5}$, p-value = 0.0352). Il miglioramento assoluto di circa 2.6 punti è inferiore a quello osservato per il modello da 1 miliardo (14 punti), il che suggerisce che modelli più grandi — che partono da un baseline già più alto, 74% contro 57% — abbiano meno margine di miglioramento.

Per Llama-3.1-8B-Instruct, la situazione è più complessa. Il modello da 8 miliardi — si noti che non esiste una versione ufficiale Llama-3.2-8B e gli autori ricorrono al predecessore Llama-3.1 — presenta un baseline anomalamente basso, 35.77% ± 6.60%, che gli autori attribuiscono a difficoltà del modello nel terminare correttamente la generazione delle espressioni regolari. LLM-JEPA raggiunge 63.57% ± 16.81% ($k = 0$, $\lambda = 2.0$, $\eta = 2\times10^{-5}$, p-value = 0.0131), con un miglioramento di circa 28 punti assoluti e una varianza molto elevata. L'ampiezza della deviazione standard (16.81%) indica una forte sensibilità

all'inizializzazione per questo modello su questo task specifico, il che rende il confronto più difficile da interpretare.

Per OLMo-2-1124-7B-Instruct, il baseline raggiunge 87.26% ± 0.27% e LLM-JEPA 87.75% ± 0.33% ($k = 2$, $\lambda = 20$, $\eta = 2\times10^{-5}$, p-value = 0.0345). La configurazione ottimale richiede un valore di λ molto elevato (20) rispetto alle configurazioni tipicamente osservate per i modelli da 1 miliardo, il che è interessante: modelli più grandi e più potenti potrebbero richiedere un segnale JEPA più intenso per trarre beneficio dalla strutturazione latente aggiuntiva, forse perché le loro rappresentazioni sono già più elaborate e richiedono una perturbazione maggiore per essere riorganizzate.

10.8 Analisi delle rappresentazioni latenti: geometria e struttura

Complementare all'analisi quantitativa delle accuratezze è l'analisi qualitativa delle strutture di rappresentazione apprese dalle due varianti di addestramento. La Figura 4 e la Figura 6 del paper mostrano proiezioni t-SNE degli embedding di Text e Code per tre configurazioni: il modello base senza fine-tuning, il modello con fine-tuning generativo standard, e il modello con LLM-JEPA.

Nel modello base, le rappresentazioni di Text e Code formano cluster separati nel piano t-SNE, con una struttura che riflette la separazione tra le due modalità linguistiche. Dopo il fine-tuning generativo standard, i cluster diventano più mescolati ma la struttura è in parte persa: le rappresentazioni si disperdono nello spazio in modo meno regolare. Con LLM-JEPA, al contrario, si osserva l'emergere di una struttura netta: le rappresentazioni di Text e Code si allineano in cluster corrispondenti, dove a ogni campione di Text ne corrisponde uno di Code geometricamente vicino nello spazio latente.

Questa osservazione è quantificata dall'analisi dei valori singolari della matrice differenza $\mathrm{Enc(Text)} - \mathrm{Enc(Code)}$. Per il modello base e per il fine-tuning generativo, i 100 valori singolari più grandi decrescono lentamente e rimangono su ordini di grandezza elevati (10^2 - 10^3). Per LLM-JEPA con $k = 1$ e con $k = 0$, i valori singolari sono di diversi ordini di grandezza inferiori (10^1 per $k = 1$, ancora inferiori per $k = 0$), indicando che la differenza $\mathrm{Enc(Text)} - \mathrm{Enc(Code)}$ è confinata in un sottospazio di dimensione molto ridotta. In altri termini, il mapping da Enc(Text) a Enc(Code) è sostanzialmente lineare e a basso rango, il che corrisponde a una struttura latente altamente regolare.

Questo risultato è verificato analiticamente tramite la regressione ai minimi quadrati: si calcola la matrice $X^* = \mathrm{argmin}_X \| \mathrm{Enc(Text)} \cdot X - \mathrm{Enc(Code)} \|^2$ e si misura il residuo $\| \mathrm{Enc(Text)} \cdot X^* - \mathrm{Enc(Code)} \|^2$. Per il modello base il residuo è 3953.11, per il fine-tuning generativo 3035.01, mentre per LLM-JEPA con $k = 1$ scende a 4.47 e con $k = 0$ a 4.04 — una riduzione di circa tre ordini di grandezza. Il mapping da Text a Code, dopo l'addestramento con LLM-JEPA, è dunque approssimato da una trasformazione lineare con errore residuo trascurabile, il che è consistente con l'ipotesi che il termine JEPA induca una struttura geometrica regolare nello spazio latente.

La congettura degli autori è che sia proprio questa regolarità geometrica — la quasi-linearità del mapping tra viste complementari — a spiegare il miglioramento di accuratezza osservato sui task generativi: un modello con uno spazio latente geometricamente strutturato può sfruttare in modo più efficiente le relazioni tra concetti, riducendo l'errore su esempi mai visti durante l'addestramento. Questa congettura rimane un'ipotesi di lavoro piuttosto che un risultato dimostrato, ma è supportata dalla coerenza tra i dati sperimentali e le previsioni teoriche.

10.9 Interpretazione probabilistica dell'iperparametro λ e regime di funzionamento ottimale

L'iperparametro λ nella perdita $\mathcal{L}$_LLM-JEPA = Σ $\mathcal{L}$_LLM + λ · d(Pred(Enc(Text)), Enc(Code)) controlla il peso relativo del segnale di allineamento latente rispetto al segnale generativo. La scelta ottimale di λ non è guidata da principi teorici semplici nel framework deterministico di LLM-JEPA, ma acquista un'interpretazione più chiara se si considera la connessione con il framework variazionale discussa nel Capitolo 2.

Nel limite in cui LLM-JEPA è interpretato come un'approssimazione deterministica di un Var-JEPA linguistico --- dove il termine d(Pred(Enc(Text)), Enc(Code)) corrisponde al termine KL deterministico $\mathcal{L}$^KL_sy nel limite di varianze posteriori tendenti a zero --- il valore ottimale di λ è determinato dalla bilancio tra la pressione di allineamento delle rappresentazioni (termine JEPA) e la pressione di generazione fedele del testo (termine LLM). Dalla prospettiva variazionale, questo bilanciamento è governato dal rapporto α^gen / α^KL_sy nell'ELBO, che controlla quanto le rappresentazioni target possono discostarsi dalla previsione del prior condizionale per seguire l'osservazione effettiva. Un λ alto corrisponde a α^gen/α^KL_sy basso --- il sistema privilegia la coerenza cross-modale delle rappresentazioni rispetto alla fedeltà generativa --- e produce uno spazio latente più strutturato ma potenzialmente meno espressivo come generatore linguistico.

I risultati sperimentali mostrano che per OLMo-2-1124-7B (modello più capace, baseling accuracy 87.26%) il valore ottimale di λ è 20 --- molto più alto dei valori 0.5-2 ottimali per i modelli da 1B. Questo pattern suggerisce che modelli più grandi e più pre-addestrati necessitano di un segnale JEPA più intenso per ristrutturare il proprio spazio latente già consolidato. Una interpretazione meccanicistica è la seguente: un modello con più parametri ha una maggiore "inerzia" dello spazio delle rappresentazioni --- le sue rappresentazioni sono stabilizzate da milioni di esempi di preaddestramento --- e richiedono un segnale di allineamento più forte per essere "spostate" verso la struttura quasi-lineare ottimale. Questo suggerisce una regola pratica: $\lambda^* \propto N_params^{\alpha}$ per qualche esponente $\alpha > 0$ da determinare empiricamente su un range più ampio di dimensioni di modello.

10.10 Generalizzazione cross-task e robustezza out-of-distribution di LLM-JEPA

I risultati empirici di Huang et al. mostrano che LLM-JEPA migliora le prestazioni su tutti i dataset testati. Ma una questione importante per la valutazione della portata del metodo riguarda la sua generalizzazione: i benefici di strutturazione latente si trasferiscono a task non visti durante il fine-tuning? E come si comporta LLM-JEPA su distribuzione di test significativamente diverse da quella di addestramento?

Il risultato di preaddestramento su cestwc/paraphrase con valutazione su Rotten Tomatoes e Yelp (Sezione 12.3) fornisce un'evidenza preliminare positiva: il modello pre-addestrato con LLM-JEPA su descrizioni fotografiche parafrasate mostra miglioramenti (+1.2 punti su Rotten Tomatoes, +0.7 su Yelp) su task di analisi del sentiment completamente diversi. Questo trasferimento cross-domain suggerisce che la strutturazione geometrica dello spazio latente indotta dall'obiettivo JEPA produce rappresentazioni più "universalmente" utili --- un indicatore di maggiore qualità intrinseca delle rappresentazioni, indipendente dal dominio specifico del fine-tuning.

Per la robustezza out-of-distribution (OOD), la riduzione della varianza delle prestazioni osservata per LLM-JEPA rispetto al baseline (deviazione standard 1.34% vs 5.32% per Llama-3.2-1B su NL-RX-SYNTH) è un indicatore indiretto: una minor varianza tra run con seed diversi suggerisce che il modello ha trovato un ottimo più stabile e meno sensibile all'inizializzazione casuale, il che è correlato con una maggiore robustezza a perturbazioni distribuzionali. Una valutazione diretta della robustezza OOD richiederebbe esperimenti su varianti sistematicamente perturbate dei dataset di test --- ad esempio, NL-RX-SYNTH con cambiamenti di vocabolario, struttura sintattica o dominio applicativo --- che non sono documentati nel paper originale ma costituiscono un'importante direzione di valutazione futura.

Capitolo 11 — Resistenza all'Overfitting: Analisi Comparativa tra Famiglie di Modelli

11.1 L'overfitting come patologia endemica del fine-tuning generativo

Chiunque abbia lavorato con il fine-tuning di modelli linguistici su dataset specializzati di dimensioni ridotte conosce la frustrazione di osservare le curve di accuratezza sul set di validazione raggiungere un picco precocemente, per poi iniziare una discesa lenta ma inesorabile mentre la perdita sul training set continua a diminuire. Questo fenomeno — l'overfitting, ovvero la memorizzazione delle specificità statistiche del set di addestramento a scapito della generalizzazione — è particolarmente insidioso per i modelli linguistici di grandi dimensioni perché la loro capacità paramétrica è enormemente superiore alla quantità di informazione contenuta nei dataset di fine-tuning tipici. Un modello con un miliardo di

parametri addestrato su qualche migliaio di esempi ha una capacità di memorizzazione che supera di ordini di grandezza quella necessaria per riprodurre esattamente il training set, e l'ottimizzazione per discesa del gradiente tende sistematicamente a sfruttare questa sovrabbondanza di capacità.

Il fatto che LLM-JEPA mostri una resistenza sistematica all'overfitting — documentata in modo rigoroso negli esperimenti del paper — è pertanto un risultato di notevole interesse pratico, oltre che teorico. Non si tratta semplicemente di un guadagno di accuratezza su un singolo punto di misura, ma di un cambiamento qualitativo nel comportamento dell'ottimizzazione che rende il modello robusto all'incremento del numero di epoche. Questa proprietà è particolarmente preziosa perché elimina in larga misura la necessità di una selezione accurata del punto di arresto anticipato (*early stopping*), che è una delle pratiche più delicate e soggettive nel fine-tuning di modelli linguistici.

11.2 Evidenza empirica della resistenza all'overfitting: il profilo temporale

La documentazione principale del fenomeno è contenuta nella Figura 1 destra del paper, che mostra l'accuratezza in funzione dell'epoca di addestramento per il dataset NL-RX-SYNTH e NL-RX-TURK con il modello Llama-3.2-1B-Instruct, confrontando baseline generativo e LLM-JEPA lungo sei epoche.

Per NL-RX-SYNTH, i valori di accuratezza del baseline lungo le sei epoche sono: 37.0%, 51.6%, 56.0%, 55.2%, 51.3%, 51.5%. Il picco è raggiunto alla terza epoca (56.0%) e il comportamento successivo è di degradazione, con l'accuratezza che scende alla quarta (55.2%), quinta (51.3%) e sesta epoca (51.5%), tornando al livello della seconda. Per LLM-JEPA, la progressione è: 51.6%, 66.6%, 70.4%, 70.9%, 71.8%, 71.8% (leggendo i valori dalla Figura 1 del paper). La crescita è sostanzialmente monotona: nessuna delle misurazioni successive al picco mostra una degradazione statisticamente significativa, e l'accuratezza alla sesta epoca è la più alta registrata.

Per NL-RX-TURK, il contrasto è ancora più netto. Il baseline mostra: 11.5%, 20.2%, 19.4%, 22.5%, 21.6%, 22.5%, con oscillazioni che rendono difficile identificare un chiaro trend dopo la seconda epoca. LLM-JEPA mostra invece: 20.2%, 27.2%, 30.9%, 31.6%, 31.8%, 30.9% — una progressione quasi monotona con leggera saturazione verso la sesta epoca ma senza degradazione marcata.

Questi dati vanno letti avendo presente che le misurazioni corrispondenti alla stessa epoca con i diversi semi fissi presentano variabilità. I valori riportati nella Figura 1 sono le medie su cinque semi, e la deviazione standard associata — particolarmente ampia per il baseline (si ricordi che per NL-RX-SYNTH il baseline ha deviazione standard di 5.32% a quattro epoche) — rende alcune delle oscillazioni apparenti statisticamente non significative. Cionondimeno, il pattern complessivo è chiaro e inequivocabile: il baseline tende a raggiungere il suo massimo intorno alle epoche tre o quattro e poi stabilizzarsi o degradare, mentre LLM-JEPA continua a migliorare o si stabilizza a un valore superiore.

11.3 Il medesimo pattern nel fine-tuning con LoRA

Il fine-tuning con adattamento a basso rango (LoRA, Low-Rank Adaptation) è diventato la tecnica standard per adattare modelli linguistici di grandi dimensioni a compiti specifici quando il budget computazionale non permette l'aggiornamento dell'intero set di parametri. Nell'approccio LoRA, i pesi del modello rimangono congelati e si addestrano matrici di aggiornamento di rango $r \ll d$, dove d è la dimensione nascosta del modello. Il numero di parametri addestrabili è proporzionale a r, e valori tipici di r vanno da 8 a 512.

Huang, LeCun e Balestriero verificano se la resistenza all'overfitting di LLM-JEPA si conserva anche in questo regime. I risultati, riportati nella Tabella 8 e nella Figura 8 del paper per il dataset NL-RX-SYNTH con Llama-3.2-1B-Instruct, mostrano che la risposta è affermativa.

La Figura 8 mostra il profilo temporale dell'accuratezza su sei epoche per il fine-tuning LoRA a rango 512. Il baseline LoRA mostra: 34.6%, 42.5%, 49.1%, 50.2%, 52.4%, 48.8% — un picco alla quinta epoca (52.4%) seguito da degradazione. LLM-JEPA con LoRA mostra: 42.5%, 55.0%, 63.5%, 72.4%, 68.9%, 72.0% — un profilo con oscillazioni più ampie ma con una tendenza crescente che raggiunge il suo massimo alla quarta epoca e non mostra un chiaro collasso successivo. La deviazione standard, indicata dalle barre di errore nella figura, è sistematicamente più alta per LLM-JEPA in questo regime LoRA rispetto al full fine-tuning, il che gli autori attribuiscono alla capacità limitata di LoRA — che aggiorna solo una frazione dei parametri — di sfruttare appieno il segnale strutturale fornito dal termine JEPA.

Un aspetto interessante segnalato dagli autori è che nelle varianti LoRA si osserva un pattern asimmetrico nella relazione tra alta deviazione standard e direzione dell'oscillazione: per LLM-JEPA le deviazioni standard elevate tendono a coincidere con i punti di minimo locale (l'accuratezza cade ma poi risale), mentre per il baseline tendono a coincidere con i picchi (l'accuratezza sale momentaneamente ma poi scende). Questo suggerisce che il termine JEPA agisce come una forza di richiamo verso un bacino di attrazione più ampio, da cui l'ottimizzazione può momentaneamente uscire ma a cui tende a tornare, mentre il baseline generativo puro può sfuggire verso regioni di alta accuratezza sul training set che però corrispondono a bassa generalizzazione.

11.4 Analisi comparativa per ranghi LoRA e famiglie di modelli

La Tabella 8 del paper riporta i risultati per ranghi LoRA crescenti — 32, 64, 128, 256, 512 — confrontando baseline e LLM-JEPA, sempre su NL-RX-SYNTH con Llama-3.2-1B-Instruct e configurazione $k = 1$, $\lambda = 1$, $\eta = 2\times10^{-5}$.

Rango 32 (percentuale di parametri addestrabili minima): baseline 6.09% ± 0.55%, LLM-JEPA 7.45% ± 1.87%. Il guadagno è di circa 1.4 punti percentuali. Il livello assoluto di accuratezza è molto basso perché a rango 32 il modello LoRA ha capacità adattativa estremamente ridotta.

Rango 64: baseline 21.09% ± 1.90%, LLM-JEPA 32.46% ± 1.26%. Il guadagno di circa 11 punti è il più ampio in termini assoluti tra tutti i ranghi, e suggerisce che, passando da

capacità molto ridotta a capacità moderata, LLM-JEPA riesce a sfruttare in modo molto più efficiente i parametri aggiuntivi disponibili.

Rango 128: baseline 34.21% ± 2.82%, LLM-JEPA 48.45% ± 3.66%. Il guadagno si mantiene intorno ai 14 punti.

Rango 256: baseline 45.57% ± 4.52%, LLM-JEPA 60.80% ± 2.31%. Il guadagno rimane di circa 15 punti, e si nota ancora una volta la riduzione della deviazione standard per LLM-JEPA rispetto al baseline.

Rango 512 (22.59% dei parametri addestrabili): baseline 50.18% ± 5.15%, LLM-JEPA 72.41% ± 2.94%. Il guadagno di oltre 22 punti è il più alto in assoluto, e il valore di LLM-JEPA con LoRA a rango 512 (72.41%) è sostanzialmente equivalente all'accuratezza del full fine-tuning con LLM-JEPA (71.46%). Al contrario, il baseline con LoRA a rango 512 (50.18%) rimane significativamente inferiore al full fine-tuning baseline (57.29%), con un divario di circa 7 punti.

Questo risultato ha implicazioni pratiche importanti: LLM-JEPA con LoRA a rango 512 raggiunge le stesse prestazioni del full fine-tuning con LLM-JEPA utilizzando solo il 22.59% dei parametri, mentre il baseline con LoRA non riesce ad avvicinarsi al full fine-tuning neanche al rango massimo considerato. In altri termini, il segnale JEPA sembra compensare almeno parzialmente la riduzione di capacità adattativa introdotta da LoRA, rendendo il modello capace di apprendere rappresentazioni di qualità superiore anche con un numero limitato di parametri addestrabili.

11.5 La struttura latente a basso rango come spiegazione meccanicistica

Perché LLM-JEPA resiste all'overfitting mentre l'addestramento generativo standard non lo fa? Una risposta puramente empirica — il termine JEPA funziona come regolarizzatore — è corretta ma non esplicativa. Per comprendere il meccanismo sottostante è necessario tornare all'analisi geometrica dello spazio latente discussa nel capitolo precedente.

Si è già osservato che dopo l'addestramento con LLM-JEPA il mapping da Enc(Text) a Enc(Code) è approssimato da una trasformazione lineare con errore residuo trascurabile (dell'ordine di 4-5, contro i 3000-4000 del modello base e del fine-tuning generativo). Questo significa che le rappresentazioni delle due viste sono vincolate a giacere in sottospazi allineati dello spazio latente, connessi da una trasformazione a basso rango.

Questa struttura ha conseguenze dirette sulla generalizzazione. Un modello che ha appreso un mapping lineare a basso rango tra viste complementari dispone implicitamente di una forma di regolarizzazione strutturale: per generalizzare correttamente su nuovi esempi, è sufficiente che il mapping rimanga coerente con la struttura lineare appresa, senza bisogno di memorizzare le specificità di ogni singolo campione. Al contrario, un modello addestrato con la sola perdita generativa non ha alcun incentivo a strutturare le sue rappresentazioni in modo da rispettare una tale coerenza, e tende a sviluppare rappresentazioni idiografiche — specifiche per ogni campione — che non trasferiscono bene su esempi nuovi.

In termini più formali, si può pensare al termine JEPA come a un vincolo implicito sulla curvatura dello spazio delle rappresentazioni nella direzione ortogonale al sottospazio a basso rango che connette le viste. Minimizzare la distanza tra Pred(Enc(Text)) ed Enc(Code) equivale a richiedere che esista una mappa predictor g tale che g(Enc(Text)) ≈ Enc(Code) per tutti i campioni del training set. Se il predictor è un'applicazione a bassa complessità (il che è incoraggiato dall'uso di pochi token di predizione k), questa condizione implica che le rappresentazioni di Text e Code debbano essere vincolate in un sottospazio di dimensione ridotta, poiché solo in questo caso una mappa semplice può connetterle in modo affidabile.

Il legame con la teoria della generalizzazione è diretto: la complessità efficace del modello — misurata dalla dimensione del sottospazio latente occupato dalle rappresentazioni — è ridotta dal termine JEPA, e secondo i classici risultati della teoria dell'apprendimento statistico una minore complessità effettiva si traduce in migliori garanzie di generalizzazione. La resistenza all'overfitting di LLM-JEPA non è pertanto un effetto casuale o specifico del dataset considerato, ma la conseguenza prevedibile della strutturazione latente operata dal termine di predizione nello spazio degli embedding.

8.8 Analisi geometrica dell'allineamento latente: decomposizione SVD e sottospazi di bassa dimensione

Il risultato empirico più illuminante documentato in LLM-JEPA riguarda la struttura geometrica dello spazio latente dopo il fine-tuning con l'obiettivo JEPA: la mappa Enc(Text) → Enc(Code) è approssimabile con una trasformazione lineare con errore residuo di circa 4.5, contro i 3000-4000 del baseline non addestrato. Questo risultato, verificato attraverso la regressione ai minimi quadrati $\min_X \|\text{Enc(Text)} \cdot X - \text{Enc(Code)}\|^2$, merita un'analisi teorica più approfondita in termini di decomposizione a valori singolari (SVD).

La decomposizione SVD della differenza $D = \text{Enc(Text)} - \text{Enc(Code)} \in \mathbb{R}^{N \times d_model}$ (dove N è il numero di coppie di test e d_model è la dimensione dell'embedding) fornisce: $D = U \Sigma V^\top$, dove $\Sigma = \text{diag}(\sigma_1, ..., \sigma_{\min(N, d_model)})$ è la matrice dei valori singolari in ordine decrescente, $U \in \mathbb{R}^{N \times N}$ e $V \in \mathbb{R}^{d_model \times d_model}$ sono matrici unitarie. La norma di Frobenius $\|D\|_F^2 = \Sigma_i \sigma_i^2$ misura la distanza quadratica totale tra le rappresentazioni delle due modalità, mentre il rango effettivo $r = \#\{i : \sigma_i > \varepsilon\}$ (con ε soglia opportuna) misura la dimensionalità del sottospazio che separa le due nuvole di punti. I risultati di Huang et al. mostrano che per LLM-JEPA i valori singolari decrescono di circa 3 ordini di grandezza rispetto al baseline: $\sigma_1(\text{LLM-JEPA})/\sigma_1(\text{baseline}) \approx 10^{-2}$, indicando che le due nuvole di punti sono quasi complanari nel sense della norma spettrale.

Una prospettiva complementare è quella dei Canonical Correlation Analysis (CCA, Hotelling 1936), che trova le direzioni di massima correlazione tra due insiemi di variabili. Le canonical correlations $\rho_1 \geq \rho_2 \geq ... \geq \rho_d$ tra Enc(Text) e Enc(Code) misurano l'allineamento degli spazi latenti lungo le direzioni più correlate. Per un allineamento perfetto (Enc(Text) = Enc(Code) · M per qualche matrice invertibile M), tutte le correlazioni canoniche sarebbero pari a 1. La quasi-linearità documentata in LLM-JEPA corrisponde a correlazioni canoniche vicine a 1 nelle prime dimensioni e

rapidamente decrescenti nelle dimensioni successive. La SVCCA (Singular Vector CCA, Raghu et al. 2017) applicata agli embeddings di LLM-JEPA fornirebbe una misura quantitativa dell'allineamento che può essere interpretata come una generalizzazione della regressione ai minimi quadrati già esplorata da Huang et al.

La connessione tra la quasi-linearità dello spazio latente e la resistenza all'overfitting può essere formalizzata attraverso la teoria del rango di Rademacher. La complessità di Rademacher di una classe di funzioni f: $\mathbb{R}^d \to \mathbb{R}^d$ è proporzionale alla norma di Frobenius della classe, che a sua volta è limitata dal rango effettivo delle rappresentazioni. Per un modello il cui mapping Text → Code nello spazio latente ha rango effettivo $r \ll d$, la complessità di generalizzazione è $O(\sqrt{(r \cdot \log(n)/n)})$ contro $O(\sqrt{(d \cdot \log(n)/n)})$ per un mapping a rango pieno. La riduzione del rango effettivo indotta da LLM-JEPA si traduce quindi formalmente in una riduzione della complessità di generalizzazione, giustificando teoricamente la migliore performance out-of-distribution osservata negli esperimenti.

8.9 Analisi teorica del predittore a pesi condivisi: bassa dimensione e capacità espressiva

L'implementazione del predittore Pred(·) attraverso k token [PRED] con pesi condivisi al modello principale è una scelta architetturale economica ma non ovvia. Comprendere le sue proprietà teoriche --- in particolare la relazione tra il numero k di token di previsione e la capacità espressiva del predittore --- è importante per guidare le scelte di iperparametri e per comprendere le limitazioni dell'approccio.

Il predittore con $k = 0$ è triviale: Pred(Enc(Text)) = Enc(Text), e l'obiettivo JEPA riduce a minimizzare d(Enc(Text), Enc(Code)) --- la distanza coseno tra le rappresentazioni delle due sequenze senza alcuna trasformazione intermedia. Questo equivale a un vincolo di vicinanza diretto nello spazio latente, analogo a una perdita contrastiva con un solo positivo per campione. Per $k > 0$, i token [PRED] si appendono alla fine di Text e vengono elaborati attraverso gli strati di self-attention del transformer. Il token [PRED]_{k} all'ultimo strato ha accesso all'intera sequenza Text attraverso la finestra causale di attenzione, e può quindi eseguire operazioni di sintesi e trasformazione dell'embedding di Text. Con k strati aggiuntivi di "processing" (in realtà k token che sfruttano la profondità L del transformer esistente, non strati aggiuntivi), la capacità espressiva del predittore è proporzionale a $k \cdot L$ operazioni di attenzione e feed-forward.

La griglia di ricerca $(k, \lambda) \in \{0,1,2,3,4\} \times \{0.5,1,2,4\}$ riportata in Figura 7 mostra un paesaggio di iperparametri irregolare: le superfici di accuratezza hanno picchi in posizioni diverse per dataset diversi, senza un pattern sistematico. Questo suggerisce che il mapping ottimale Text → Code è diverso per task diversi: per NL-RX-SYNTH (dove la relazione testo-regex è altamente strutturata), $k = 1$ e $\lambda = 1$ sono ottimali per Llama-3.2-1B; per GSM8K (dove il mapping è più asimmetrico), $k = 4$ e $\lambda = 0.5$ sono preferibili. Una interpretazione teorica di questo pattern è la seguente: il numero ottimale k di passi di "planning" del predittore è proporzionale alla complessità della trasformazione necessaria per allineare le rappresentazioni delle due viste, misurata dal

rango effettivo della matrice di allineamento ottimale. Per viste semanticamente vicine (testo e codice per regex), il mapping è quasi-lineare e richiede pochi passi (k = 1); per viste più distanti (testo del problema e risposta numerica in GSM8K), il mapping richiede più elaborazione (k = 4).

11.6 L'ablazione del termine generativo: quando il solo JEPA non basta

Un risultato controintuitivo ma metodologicamente importante è quello riportato nella Tabella 10 del paper, che studia l'effetto della variazione del rapporto γ/λ tra il peso del termine generativo e il peso del termine JEPA. Nella formulazione estesa della perdita:

$$\mathcal{L}_{\text{LLM-JEPA}} = \gamma \cdot \Sigma_{\ell=2}^{L} \mathcal{L}_{\text{LLM}}(\text{Text}_{1:\ell-1}, \text{Text}_\ell) + \lambda \cdot d(\text{Pred}(\text{Enc}(\text{Text})), \text{Enc}(\text{Code}))$$

il parametro γ controlla l'intensità del termine generativo, mentre λ controlla quella del termine JEPA. Variando il rapporto γ/λ con il vincolo max(γ, λ) = 1 — che mantiene costante la scala del gradiente complessivo — si ottengono configurazioni che enfatizzano progressivamente uno dei due termini a scapito dell'altro.

I risultati mostrano un comportamento asimmetrico e rivelatore. Per γ/λ = 1.0 (peso uguale per entrambi i termini, γ = λ = 1), l'accuratezza è 70.42% ± 2.36% — il caso ottimale. Per γ/λ = 10 (termine generativo dieci volte più intenso del termine JEPA, con γ = 1, λ = 0.1), l'accuratezza scende a 67.52% ± 1.45%. Per γ/λ = 100 (γ = 1, λ = 0.01), scende ulteriormente a 66.83% ± 3.89%. Per γ/λ → ∞ (λ = 0, solo il termine generativo), si riottiene il baseline con 57.29% ± 5.32%.

La direzione opposta è ancora più istruttiva. Per γ/λ = 0.1 (γ = 0.1, λ = 1), l'accuratezza è 45.80% ± 5.04% — nettamente inferiore al baseline puro. Per γ/λ = 0.01 (γ = 0.01, λ = 1), l'accuratezza scende a 1.38% ± 0.06%. Per γ/λ = 0 (γ = 0, solo il termine JEPA, nessun contributo generativo), l'accuratezza è 0.00% ± 0.00%: il modello produce esclusivamente output vuoti.

Questo risultato chiarisce in modo definitivo il ruolo dei due componenti. Il termine generativo $\mathcal{L}_{\text{LLM}}$ è indispensabile: senza di esso il modello non apprende a produrre output coerenti, poiché il termine JEPA opera esclusivamente sullo spazio delle rappresentazioni latenti e non fornisce alcun segnale sulla forma della sequenza da generare. Il termine JEPA è invece un regolarizzatore: migliora le rappresentazioni latenti in modo da favorire la generalizzazione, ma non può sostituire il segnale primario di addestramento. La metafora più appropriata è quella di un vincolo geometrico imposto su uno spazio che deve essere prima abitato dalla soluzione e poi organizzato al suo interno: senza la popolazione dello spazio da parte del termine generativo, il vincolo geometrico non ha nulla su cui agire.

Questa interpretazione è coerente con il risultato osservato nella Figura 3 del paper, che mostra come la perdita generativa $\mathcal{L}_{\text{LLM}}$ rimanga sostanzialmente identica nelle varianti baseline e LLM-JEPA durante l'addestramento: il termine JEPA non migliora la

minimizzazione della perdita generativa, ma struttura le rappresentazioni in modo che la stessa minimizzazione dia origine a caratteristiche più generalizzabili.

11.7 Confronto qualitativo tra famiglie di modelli: similitudini e differenze

Un aspetto rilevante che emerge dall'esame comparativo dei risultati tra le quattro famiglie di modelli è che, pur mostrando tutte una resistenza all'overfitting con LLM-JEPA, le peculiarità architetturali e di addestramento di ciascuna famiglia si riflettono nella forma specifica del comportamento.

Per Gemma-2-2b-it, il modello più grande della serie testata (2.2 miliardi di parametri nominali), la configurazione ottimale di LLM-JEPA richiede $k = 4$ e $\lambda = 2$ — valori di k elevati e di λ moderatamente alti rispetto alla configurazione di Llama. Questo suggerisce che un modello più capace richiede un predictor più complesso per strutturare efficacemente le sue rappresentazioni latenti, poiché lo spazio in cui tali rappresentazioni vivono è più ricco e articolato.

Per OpenELM-1_1B-Instruct, un modello progettato da Apple con tecniche di allocazione dei parametri specifiche per ciascuno strato (*layer-wise parameter allocation*), la configurazione ottimale richiede $k = 3$ e $\lambda = 4$ — il valore più alto di λ tra tutte le famiglie testate. L'elevato peso del termine JEPA necessario per ottenere il massimo beneficio potrebbe riflettere il fatto che l'architettura non uniforme di OpenELM produce spazi latenti con strutture geometriche diverse da strato a strato, richiedendo un segnale di regolarizzazione più intenso per indurre coerenza globale.

Per OLMo-2-0425-1B-Instruct, la configurazione ottimale con $k = 0$ e $\lambda = 2$ — predictor banale, cioè identità — suggerisce che per questo modello la semplice imposizione della prossimità tra le rappresentazioni delle viste (senza trasformazione del predictor) è sufficiente per ottenere il beneficio strutturale. Ciò può riflettere il fatto che le rappresentazioni prodotte da OLMo sono già naturalmente più allineate tra viste diverse grazie alle sue procedure di addestramento, e richiedono solo una regolarizzazione lieve piuttosto che una ristrutturazione profonda.

Queste differenze tra famiglie non invalidano la generalità del risultato — LLM-JEPA migliora la generalizzazione per tutti i modelli testati — ma indicano che la configurazione ottimale degli iperparametri (k, λ) dipende in modo non banale dall'architettura specifica, e che una comprensione più profonda di questa dipendenza richiederebbe un'analisi sistematica delle proprietà geometriche degli spazi latenti dei diversi modelli, al momento non disponibile nella letteratura.

11.8 La regolarizzazione geometrica come compressione strutturale: teoria MDL

La resistenza all'overfitting di LLM-JEPA può essere analizzata attraverso la teoria Minimum Description Length (MDL, Rissanen 1978). MDL propone che il modello

ottimale è quello che permette la descrizione più compressa dei dati. La perdita JEPA aggiunge un termine di descrizione supplementare: il costo di codificare il mapping Enc(Text) → Enc(Code) in modo compresso. Se questo mapping è quasi-lineare a basso rango — come documentato dall'analisi SVD — allora il suo costo di codifica è O(r * d_model) bit dove r è il rango effettivo, molto inferiore al costo O(d_model^2) di un mapping generico. Il termine JEPA nella perdita spinge il modello verso mapping a basso rango che sono più comprimibili e, secondo MDL, più generalizzabili. La complessità stocastica del mapping è proporzionale a $\Sigma_i \log(\sigma_i(D))$ dove σ_i sono i valori singolari della differenza D = Enc(Text) − Enc(Code). Poiché LLM-JEPA riduce questi valori singolari di circa 3 ordini di grandezza, riduce la complessità stocastica di circa 10 bit per dimensione — una riduzione che si traduce direttamente in migliore generalizzazione.

11.9 Confronto con weight decay, dropout e label smoothing

LLM-JEPA produce regolarizzazione implicita attraverso la strutturazione geometrica dello spazio latente. Il weight decay penalizza la norma dei pesi (regolarizzazione parametrica), mentre LLM-JEPA vincola la struttura delle rappresentazioni apprese (regolarizzazione funzionale). Per task dove la struttura delle rappresentazioni è più critica della magnitudine dei pesi — come nel ragionamento simbolico e nella generazione di codice — LLM-JEPA è teoricamente superiore. Il dropout introduce rumore nelle attivazioni forzando ridondanza robusta, mentre LLM-JEPA impone coerenza cross-modale strutturale. I due meccanismi sono complementari e la loro combinazione potrebbe produrre effetti sinergici, poiché agiscono su aspetti diversi del processo di generalizzazione.

Capitolo 12 — Implicazioni per il Pre-addestramento e il Fine-tuning con Obiettivi JEPA

12.1 Due regimi, una stessa logica

Il lavoro di Huang, LeCun e Balestriero si concentra prevalentemente sul fine-tuning — ovvero sull'adattamento di un modello già pre-addestrato su corpora generici a un compito specifico mediante ulteriore ottimizzazione su un dataset specializzato. Questa è la modalità operativa più comune nell'era dei grandi modelli linguistici: partire da un modello fondazionale come Llama o Gemma e adattarlo a un dominio o a un formato di output particolare è molto più efficiente, in termini di risorse computazionali, che addestrare un modello da zero per ogni applicazione. I risultati discussi nei capitoli precedenti appartengono tutti a questo regime.

Tuttavia, la domanda più profonda — e quella con le implicazioni di più lungo periodo — riguarda l'applicabilità del termine JEPA al pre-addestramento, ovvero alla fase in cui il

modello costruisce le sue rappresentazioni fondamentali del linguaggio a partire da corpus di testo grezzo. Se LLM-JEPA funziona come regolarizzatore nel fine-tuning, le sue proprietà di strutturazione latente dovrebbero manifestarsi ancora più nettamente nel pre-addestramento, dove le rappresentazioni sono costruite ex novo e possono essere organizzate dall'inizio secondo la geometria indotta dall'obiettivo JEPA. Il paper esplora questa ipotesi in due esperimenti preliminari distinti, i cui risultati sono promettenti pur nella loro necessaria incompletezza.

12.2 Pre-addestramento su NL-RX-SYNTH da inizializzazione casuale

Il primo esperimento di pre-addestramento consiste nell'addestrare il modello Llama-3.2-1B-Instruct a partire da pesi inizializzati casualmente — senza dunque sfruttare le rappresentazioni già apprese durante il pre-addestramento standard su corpora di larga scala — sul dataset NL-RX-SYNTH. L'obiettivo è verificare se LLM-JEPA sia in grado di guidare l'apprendimento di rappresentazioni utili anche in assenza della conoscenza linguistica pregressa tipicamente fornita dal pre-addestramento.

Il confronto diretto con il fine-tuning è metodologicamente improprio, perché i punti di partenza dei due esperimenti sono radicalmente diversi. È pertanto più corretto leggere i risultati del pre-addestramento come una valutazione della capacità di LLM-JEPA di strutturare l'apprendimento a partire da zero, piuttosto che come una competizione con il fine-tuning.

La metrica adottata per questo esperimento è necessariamente diversa da quella usata nel fine-tuning. Poiché il dataset NL-RX-SYNTH è di dimensioni limitate e il modello pre-addestrato da zero non ha appreso a terminare correttamente le sequenze generate, gli autori adottano un criterio più permissivo: una generazione è considerata corretta se inizia con la sequenza target (*startswith criterion*), indipendentemente dal fatto che termini correttamente.

Con questa metrica, il baseline generativo raggiunge 54.38% ± 1.70% su cinque semi, mentre LLM-JEPA con la configurazione ottimale ($\lambda = 2$, $k = 3$, $\eta = 8\times10^{-5}$, ereditata dagli esperimenti di fine-tuning) raggiunge 60.59% ± 1.01%. Il miglioramento assoluto è di circa 6.2 punti percentuali, con p-value = 2.94×10^{-4} — statisticamente molto significativo. Si noti che anche la deviazione standard si riduce, da 1.70% a 1.01%, confermando il pattern di stabilizzazione già osservato nel fine-tuning.

Questo risultato dimostra che LLM-JEPA non si limita a sfruttare le rappresentazioni già costruite dal pre-addestramento per organizzarle meglio durante il fine-tuning, ma è capace di guidare la costruzione di rappresentazioni strutturate anche a partire da pesi casuali. Il termine JEPA possiede pertanto una capacità strutturante intrinseca che non dipende dalla qualità delle rappresentazioni iniziali, ma si manifesta già nelle prime fasi dell'apprendimento.

12.3 Pre-addestramento su parafrasi e trasferimento a task eterogenei

Il secondo esperimento di pre-addestramento è strutturalmente diverso e methodologicamente più sofisticato. In questo caso, il dataset di pre-addestramento è cestwc/paraphrase, una raccolta di gruppi di cinque parafrasi: ogni campione è un insieme di cinque riformulazioni di uno stesso enunciato. Le viste per il termine JEPA sono costruite all'interno di ogni gruppo: la i-esima parafrasi è usata per predire la (i+1)-esima, formando una catena di coppie con la medesima struttura sottostante ma formulazioni superficiali diverse.

L'obiettivo del termine JEPA in questo contesto è incentivare il modello a costruire rappresentazioni latenti che catturino il contenuto semantico invariante tra le parafrasi, ignorando le variazioni superficiali di formulazione. Questo corrisponde esattamente alla proprietà desiderata per un modello linguistico general-purpose: rappresentare il significato piuttosto che la forma.

Il modello è pre-addestrato per quattro epoche sul dataset delle parafrasi, e poi valutato su due task di classificazione del sentiment — Rotten Tomatoes (classificazione binaria: positivo/negativo) e Yelp (classificazione a cinque classi) — mediante un'ulteriore fase di fine-tuning di una sola epoca. Il fine-tuning per la valutazione non utilizza il termine JEPA, così che il vantaggio osservato possa essere attribuito esclusivamente alla fase di pre-addestramento.

Per Rotten Tomatoes, il baseline di pre-addestramento (solo $\mathcal{L}$_LLM) raggiunge 56.57% ± 1.66% nel fine-tuning successivo, mentre LLM-JEPA raggiunge 57.76% ± 1.33% ($\lambda = 0.5$, $k = 2$, p-value = 7.38×10^{-4}). Per Yelp, i valori corrispondenti sono 26.46% ± 0.92% e 27.15% ± 0.93% (p-value = 1.00×10^{-3}).

I miglioramenti assoluti — circa 1.2 punti su Rotten Tomatoes e 0.7 punti su Yelp — sono modesti in termini assoluti, ma due considerazioni li rendono significativi. Prima di tutto, non esiste alcun legame diretto tra il dominio del pre-addestramento (descrizioni fotografiche parafrasate nel dataset cestwc/paraphrase) e i task di valutazione (critiche cinematografiche su Rotten Tomatoes, recensioni di ristoranti su Yelp): il trasferimento è puramente latente, mediato dalla struttura delle rappresentazioni apprese. In secondo luogo, il termine JEPA applicato durante il solo pre-addestramento — senza ulteriore utilizzo durante il fine-tuning — produce un miglioramento che persiste e si manifesta su task completamente diversi, il che suggerisce che il beneficio operi al livello della qualità intrinseca delle rappresentazioni latenti piuttosto che come adattamento specifico al dominio del pre-addestramento.

La Tabella 7 del paper documenta le capacità generative del modello pre-addestrato con LLM-JEPA: quando vengono forniti i primi token di una sequenza del dataset di parafrasi, il modello genera completamenti che sono semanticamente plausibili anche se non coincidono letteralmente con le versioni di riferimento. Questa evidenza qualitativa conferma che il pre-addestramento con LLM-JEPA non compromette le capacità generative fondamentali del modello, coerentemente con quanto osservato nel fine-tuning.

12.4 Fine-tuning versus pre-addestramento: un quadro comparativo

Mettendo a confronto i due regimi operativi, emergono alcune differenze strutturali che vale la pena analizzare esplicitamente, poiché hanno implicazioni dirette per la progettazione di sistemi basati su LLM-JEPA.

Nel fine-tuning, il modello parte da una configurazione già ricca di conoscenza linguistica acquisita durante il pre-addestramento su corpora generici. Il termine JEPA opera in questo contesto come un vincolo geometrico sullo spazio delle rappresentazioni pre-esistenti, spingendo quelle delle due viste a collocarsi in regioni coerenti e a basso rango. L'effetto è immediato e pronunciato già nelle prime epoche, come documentato dai grafici temporali del capitolo precedente. La dimensione del dataset è relativamente piccola (poche migliaia di esempi nei dataset considerati), il che rende il problema dell'overfitting particolarmente acuto e i benefici della regolarizzazione JEPA immediatamente visibili.

Nel pre-addestramento, il modello costruisce le sue rappresentazioni ex novo a partire da pesi casuali. Il termine JEPA ha in questo caso un ruolo formativo piuttosto che correttivo: non riorganizza rappresentazioni già esistenti ma guida la formazione di rappresentazioni che abbiano fin dall'inizio le proprietà geometriche desiderate. L'effetto benefico del pre-addestramento JEPA è dunque più silente ma potenzialmente più profondo: si manifesta come una migliore qualità delle rappresentazioni fondamentali del modello, che trasferisce su task eterogenei anche senza ulteriore utilizzo del termine JEPA nel fine-tuning.

Una differenza pratica importante riguarda i requisiti sul dataset. Nel fine-tuning, la struttura di viste complementari (testo-codice, domanda-risposta) è tipicamente già presente nel dataset specifico del task. Nel pre-addestramento su corpora generici, non esiste una tale struttura naturale: i corpus di testo grezzo come Common Crawl o Wikipedia non contengono coppie di sequenze che siano genuinamente due proiezioni della stessa conoscenza sottostante. Il dataset di parafrasi è un caso fortunato e raro; per la maggior parte dei corpus di pre-addestramento, la costruzione di viste complementari richiederebbe una procedura di augmentation ad hoc.

12.5 Il problema delle viste nel pre-addestramento generale

La limitazione più fondamentale di LLM-JEPA nel regime di pre-addestramento è, come riconosciuto esplicitamente dagli autori, la necessità di viste non banali. In visione artificiale, le viste per I-JEPA sono ottenute mascherando regioni diverse della stessa immagine: la struttura spaziale dell'immagine fornisce naturalmente un meccanismo per costruire viste complementari (la regione di contesto e la regione target), e la distribuzione delle maschere può essere modulata per controllare la difficoltà del compito predittivo. Il requisito di viste non banali è soddisfatto automaticamente per quasi qualsiasi immagine.

Per il testo, la situazione è fondamentalmente diversa. La struttura linguistica del testo non offre un analogo diretto della struttura spaziale dell'immagine: la masking di token, come in BERT o in altri approcci di masked language modeling, produce viste che sono parziali dello stesso testo piuttosto che proiezioni di un contenuto comune — e questo è esattamente il tipo di obiettivo nello spazio degli input che LeCun critica come sub-ottimale per l'apprendimento di rappresentazioni astratte.

Le coppie (testo, codice), (domanda, risposta), (testo, parafrasi) funzionano come viste non banali perché sono due formulazioni distinte della stessa conoscenza: la corrispondenza non è superficiale (come tra due frammenti dello stesso testo) ma semantica (tra due modalità di espressione dello stesso contenuto). Questa condizione è soddisfatta per dataset specializzati costruiti appositamente, ma non è facilmente estendibile a corpus generali.

Sviluppare un meccanismo che generi automaticamente viste non banali a partire da testi arbitrari — un analogo linguistico delle tecniche di augmentation dei dati in visione — è il problema aperto centrale per il progresso di LLM-JEPA verso il pre-addestramento su scala industriale. Alcune direzioni sono già delineabili a partire dalla struttura del problema: le parafrasi generate da un modello diverso, le traduzioni in un'altra lingua, i riassunti o le espansioni di un testo, le riformulazioni in registro formale o informale potrebbero tutti fungere da viste complementari. Ciascuna di queste opzioni introduce però complessità operativa e potenziali bias nel segnale di supervisione che richiedono studio accurato.

12.6 Il loss dropout come ponte verso il pre-addestramento scalabile

La tecnica del loss dropout discussa nel capitolo precedente acquista un significato strategico particolare alla luce della questione della scalabilità verso il pre-addestramento. Come già analizzato, il loss dropout riduce il costo computazionale per epoca da $2 \cdot C_{base}$ a $(2-\alpha) \cdot C_{base}$, rendendo il overhead computazionale di LLM-JEPA progressivamente più gestibile all'aumentare del tasso di dropout.

Per tassi di dropout elevati, come LD = 0.75, il costo per epoca è solo $1.25 \cdot C_{base}$ — un overhead del 25% rispetto all'addestramento generativo puro. Questo overhead è sostanzialmente inferiore alla tipica variabilità nei costi di addestramento dovuta a fattori come la dimensione del batch, la lunghezza delle sequenze o la complessità del dataset, e può essere considerato trascurabile in un regime di pre-addestramento su larga scala dove i costi complessivi sono dominati da fattori di infrastructure.

Gli autori sottolineano esplicitamente questa prospettiva, indicando che la combinazione della maschera di attenzione personalizzata (che riduce il numero di passaggi da tre a due) con il loss dropout aggressivo (che elimina il secondo passaggio in una frazione α dei mini-batch) offre "positive perspectives to further scale LLM-JEPA to full scale pretraining with minimal computational overhead". Non si tratta di una promessa verificata sperimentalmente, ma di una proiezione ragionevole basata sui risultati sul fine-tuning e sui due esperimenti preliminari di pre-addestramento.

12.9 Teoria formale del trasferimento rappresentazionale: distanza di rappresentazione e transferability

Il preaddestramento con LLM-JEPA migliora le prestazioni di fine-tuning su task diversi da quello di preaddestramento, come documentato dagli esperimenti con le parafrasi (Sezione 12.3). Comprendere formalmente perché e quando questo trasferimento

avviene richiede una teoria del trasferimento rappresentazionale che vada al di là delle osservazioni empiriche.

La teoria della task similarity (Zamir et al. 2018) fornisce un framework per misurare la "distanza" tra task in termini di trasferibilità delle rappresentazioni. Due task τ_1 e τ_2 sono vicini se le rappresentazioni ottimali per τ_1 sono anche buone approssimazioni delle rappresentazioni ottimali per τ_2. Formalmente, la distanza di trasferimento $D(\tau_1, \tau_2)$ può essere misurata come il costo di adattare la soluzione ottimale di τ_1 per risolvere τ_2 --- tipicamente la norma della differenza tra le feature ottimali. Per LLM-JEPA, il task di preaddestramento (allineamento testo-codice) e i task di fine-tuning (classification, SQL generation, regex generation) hanno distanze di trasferimento diverse: task che richiedono comprensione strutturale del linguaggio formale (regex, SQL) sono vicini al task di preaddestramento JEPA, mentre task di comprensione del linguaggio naturale puro (sentiment classification) sono più distanti ma beneficiano comunque della migliore struttura delle rappresentazioni.

Un approccio alternativo è il Linear CKA (Centered Kernel Alignment, Kornblith et al. 2019), che misura la similarità tra le rappresentazioni di due modelli (o dello stesso modello in due fasi di addestramento) calcolando la correlazione tra le matrici di kernel delle rispettive feature maps. Un CKA alto tra le rappresentazioni intermedie di un modello preaddestrato con LLM-JEPA e quelle del modello ottimale per il task target indica un buon trasferimento atteso. L'analisi CKA delle rappresentazioni di LLM-JEPA rispetto al baseline non è documentata nel paper originale, ma potrebbe fornire insight meccanistici sul perché certi task beneficiano più di altri dall'obiettivo JEPA.

12.10 Curriculum di preaddestramento per LLM-JEPA: dalla scelta delle viste all'ordinamento dei dati

La qualità del preaddestramento con LLM-JEPA dipende non solo dalla perdita utilizzata ma anche dalla sequenza in cui i dati vengono presentati durante l'addestramento. La teoria del curriculum learning (Bengio et al. 2009) suggerisce che l'ordinamento dei dati può avere un impatto significativo sulla qualità delle rappresentazioni apprese e sulla velocità di convergenza. Per LLM-JEPA, questa questione si pone su due livelli: l'ordinamento delle coppie di viste e la progressione della difficoltà del task di allineamento.

Per l'ordinamento delle coppie di viste, una strategia di curriculum naturale è quella di iniziare con coppie di viste semanticamente vicine (dove il mapping Text → Code nello spazio latente è quasi identitario) e progredire verso coppie semanticamente distanti (dove il mapping richiede elaborazione più profonda). Nelle coppie (testo, espressione regolare), per esempio, descrizioni semplici di pattern di base (e.g., "stringhe che contengono solo lettere") corrispondo a espressioni regolari compatte ([a-zA-Z]+), mentre descrizioni complesse con quantificatori e alternative nidificate corrispondono a espressioni regolari elaborate. Un curriculum che presenti prima le descrizioni semplici potrebbe aiutare il predittore JEPA a imparare prima le trasformazioni basilari e poi le composizioni più complesse.

Per la progressione del task di allineamento, il valore di λ potrebbe essere aumentato progressivamente durante il preaddestramento: iniziare con λ piccolo (segnale JEPA debole, il modello preserva le capacità generative base) e aumentarlo gradualmente (segnale JEPA crescente, il modello ristruttura lo spazio latente). Questa strategia di lambda annealing è analoga al KL annealing usato nei VAE per prevenire il posterior collapse: nei primi passi di addestramento, il sistema impara a generare testo fluido senza la pressione strutturante JEPA; solo successivamente, quando le capacità generative sono consolidate, viene introdotto il segnale di allineamento cross-modale. Il loss dropout può essere visto come una versione stocastica di questo curriculum: con α decrescente nel tempo (da alto a basso), il segnale JEPA viene presentato con frequenza crescente, simulando un annealing di λ effettivo.

12.7 Confronto con approcci alternativi di regolarizzazione dello spazio latente

LLM-JEPA non è il primo tentativo di introdurre vincoli sull'embedding space durante l'addestramento di modelli linguistici. Il paper cita due lavori precedenti che esplorano direzioni correlate, ed è utile confrontarli esplicitamente per collocare il contributo di LLM-JEPA nel contesto più ampio della letteratura.

Il primo è SimCSE (Gao et al., 2021), che sfrutta il dropout come meccanismo per generare viste diverse della stessa sequenza — due passaggi in avanti della stessa sequenza con maschere di dropout diverse producono rappresentazioni che vengono trattate come positive nel quadro del contrastive learning. SimCSE produce rappresentazioni di frase di qualità molto elevata, ma non mantiene le capacità generative del modello: è progettato per produrre embeddings di frase, non per addestramento autoregressivo. LLM-JEPA differisce criticamente perché mantiene intatte le capacità generative, operando come addizione alla perdita generativa piuttosto che come sostituzione.

Il secondo lavoro rilevante citato è Large Concept Models (Barrault et al., 2024), che propone di addestrare modelli linguistici in uno spazio di rappresentazioni di frasi (sentence embedding space) piuttosto che nel tradizionale spazio dei token. Questo approccio impone vincoli strutturali più forti e richiede modifiche architetturali sostanziali. LLM-JEPA è invece agnostico rispetto all'architettura del modello e si applica a qualsiasi LLM esistente senza modifiche strutturali.

Una terza direzione correlata, non discussa nel paper ma rilevante per il contesto teorico, è quella della distillazione del conhecimento (*knowledge distillation*) nello spazio latente, dove un modello studente apprende a riprodurre le rappresentazioni interne di un modello insegnante. La differenza fondamentale con LLM-JEPA è che la distillazione richiede l'esistenza di un modello insegnante pre-addestrato e opera una trasmissione di conoscenza unidirezionale, mentre LLM-JEPA impone una struttura di coerenza tra viste senza richiedere alcun modello di riferimento esterno.

12.8 Sintesi: un'architettura di addestramento con proprietà duali

Giunti al termine della Parte II, è possibile formulare una sintesi delle proprietà fondamentali di LLM-JEPA come architettura di addestramento. Due proprietà duali emergono con particolare chiarezza dall'analisi complessiva degli esperimenti.

La prima proprietà è la complementarità tra il termine generativo e il termine JEPA. Come dimostrato dall'ablazione del parametro γ, i due termini non sono intercambiabili né il secondo può sostituire il primo: il termine generativo è necessario per guidare l'apprendimento della struttura sintattica e semantica del linguaggio, mentre il termine JEPA è necessario per organizzare le rappresentazioni latenti in modo da favorire la generalizzazione. I due segnali operano su livelli diversi — uno nello spazio degli input (la sequenza di token), l'altro nello spazio degli embedding — e si completano anziché interferire.

La seconda proprietà è la scalabilità del beneficio attraverso regimi e architetture eterogenei. LLM-JEPA migliora le prestazioni in modo statisticamente significativo per tutti i modelli testati (quattro famiglie con taglie da 1 a 8 miliardi di parametri), per tutti i dataset considerati (traduzione testo-codice, ragionamento matematico, parsing SQL, comprensione del linguaggio), e in entrambi i regimi operativi (fine-tuning e pre-addestramento). Questa pervasività suggerisce che il beneficio non è specifico di particolari scelte architetturali o di addestramento, ma riflette una proprietà generale della struttura geometrica degli spazi latenti dei modelli linguistici.

Queste due proprietà, considerate congiuntamente, configurano LLM-JEPA come un principio di addestramento piuttosto che come una tecnica specifica: un modo di pensare all'addestramento dei modelli linguistici che integra il segnale predittivo nello spazio degli embedding con il segnale generativo nello spazio degli input, traendo beneficio dalla struttura di entrambi. Le implicazioni di questo principio per l'evoluzione dell'architettura dei modelli fondazionali sono l'oggetto della riflessione critica che occuperà la Parte IV di questa monografia.

Capitolo 13 — Il Radicamento Simbolico nei Modelli Linguistici di Grande Scala: Il Problema e le Soluzioni Emergenti

13.1 La Radice Epistemica del Problema

Qualunque riflessione sulla cognizione artificiale che aspiri a essere non banale deve prima confrontarsi con una questione che la tradizione analitica della filosofia della mente chiama il *problema del radicamento simbolico* (*symbol grounding problem*). La formulazione canonica, dovuta a Harnad (1990), articola il nodo teorico con notevole precisione: un sistema di simboli — intesi come token astratti combinabili secondo regole sintattiche — è di per sé

semanticamente vuoto finché i propri elementi primitivi non siano ancorati a qualcosa che stia al di fuori del sistema medesimo. Detto altrimenti, il significato non emerge dalla ricombinazione di simboli, per quanto sofisticata e gerarchicamente strutturata essa sia; esso presuppone un aggancio con il mondo, con quelle entità, proprietà e relazioni che i simboli pretendono di rappresentare.

L'analogia classica con la stanza di Searle non è casuale: un sistema che manipola sequenze di caratteri cinesi secondo regole sintattiche prestabilite si comporta esteriormente come un parlante nativo senza che nessuna comprensione semantica sorga dall'interno del processo. I Grandi Modelli Linguistici — nella misura in cui sono addestrati a prevedere token successivi sulla base di distribuzioni statistiche estratte da enormi corpora — si trovano esattamente in questa posizione. La loro straordinaria competenza sintattica e distribuzionale non è di per sé garanzia che le rappresentazioni interne abbiano un referente stabile e non arbitrario nel mondo extralinguistico.

La questione non è accademica. Mumuni e Mumuni (2025) la collocano tra i quattro ostacoli fondamentali al conseguimento dell'intelligenza artificiale generale nei sistemi basati su LLM, al fianco di embodiment, causalità e memoria. Le capacità generaliste già dimostrate da questi modelli — ragionamento commonsense, pianificazione, dialogo aperto — rimangono *superficiali e fragili* proprio perché le rappresentazioni apprese tendono a catturare pattern associativi senza ancorarsi ai meccanismi causali e percettivi che sottendono la cognizione biologica. Questo capitolo ricostruisce in dettaglio la struttura del problema e le principali famiglie di approcci che la ricerca recente ha prodotto per affrontarlo.

13.2 Struttura Formale del Sistema Simbolico

Prima di analizzare le soluzioni, conviene precisare cosa si intende per *sistema simbolico* in senso proprio. Newell e Simon, nella loro ipotesi del sistema fisico di simboli, sostengono che simboli fisici siano necessari e sufficienti per l'intelligenza; la tradizione cognitivista costruisce su questa tesi una concezione computazionale della mente in cui ogni processo cognitivo è riducibile a trasformazioni sintattiche su strutture di dati discrete.

Formalmente, un sistema simbolico S è definito da una tripla $\langle \mathcal{A}, \mathcal{R}, \mathcal{F} \rangle$ dove:

- $\mathcal{A} = \{a_1, a_2, \ldots, a \}$ è l'alfabeto di simboli primitivi (atomi);
- $\mathcal{R} \subseteq \mathcal{A}^k$ per vari k è l'insieme delle relazioni sintattiche che regolano la composizione;
- $\mathcal{F}$ è l'insieme delle funzioni di trasformazione che mappano strutture simboliche in strutture simboliche.

Il problema del radicamento emerge precisamente perché la tripla $\langle \mathcal{A}, \mathcal{R}, \mathcal{F} \rangle$ è definita in termini puramente intrasistematici: l'assegnazione di significato M: $\mathcal{A} \rightarrow \mathcal{W}$ — dove $\mathcal{W}$ è un qualche spazio semantico che include entità, proprietà e relazioni del mondo — non è derivabile dall'interno del sistema, ma deve essere introdotta dall'esterno. La domanda centrale è dunque: come costruire tale funzione M in modo che sia stabile, composizionale e trasferibile a nuovi contesti?

Nei modelli linguistici, la funzione di codifica Enc: $\mathcal{T} \to \mathbb{R}^d$ che mappa token o sequenze di token in vettori dello spazio di rappresentazione produce un'approssimazione implicita di M. Tuttavia, come si dimostrerà, questa approssimazione è vincolata dai dati di addestramento e non garantisce né la correttezza del radicamento né la sua trasferibilità fuori distribuzione.

13.3 Il Radicamento come Funzione di Bridging

Mumuni e Mumuni propongono di concettualizzare il radicamento come una funzione di *bridging* tra lo spazio digitale delle rappresentazioni interne e lo spazio fisico del mondo reale. L'obiettivo è stabilire la corrispondenza corretta tra le rappresentazioni astratte catturate nei modelli di IA e il mondo fisico che esse cercano di rappresentare; così facendo si consente ai sistemi artificiali di dare senso agli input ambientali, potenziando la consapevolezza situazionale e il comportamento contestualmente appropriato.

Più precisamente, sia $\mathcal{L}$ lo spazio latente del modello e sia $\mathcal{W}$ lo spazio semantico-referenziale del mondo. Una funzione di radicamento G è una mappa:

$G: \mathcal{L} \to \mathcal{P}(\mathcal{W})$

che associa a ogni vettore latente $l \in \mathcal{L}$ un insieme di possibili referenti $G(l) \subseteq \mathcal{W}$. Il radicamento è *corretto* se G(Enc(s)) contiene effettivamente il referente del simbolo s nel contesto considerato; è *composizionale* se $G(Enc(s_1 \circ s_2))$ è funzione sistematica di $G(Enc(s_1))$ e $G(Enc(s_2))$; è *invariante* se G(Enc(s)) rimane stabile attraverso trasformazioni superficiali di s che non alterano il contenuto semantico.

Queste tre proprietà raramente coesistono nei sistemi attuali. La correttezza è spesso violata dall'allucinazione, fenomeno in cui il modello assegna rappresentazioni plausibili a referenti inesistenti o errati. La composizionalità fallisce nei casi di composizione non lineare o di polisemia contestuale. L'invarianza è compromessa dalla dipendenza distribuzionale: simboli identici in contesti distributivi diversi possono ricevere rappresentazioni latenti sostanzialmente diverse.

13.4 Approcci Classici e Loro Limitazioni

La tradizione più antica nel radicamento artificiale impiega rappresentazioni esplicite con regole fisse e ontologie per descrivere le relazioni e le proprietà delle entità coinvolte. Le tecniche di manipolazione simbolica basate su *variable binding* — ovvero la specifica di come variabili libere in espressioni simboliche vengano legate a valori concreti — e su regole logiche offrono trasparenza e interpretabilità, ma al prezzo di una rigidità strutturale che le rende inadatte a scenari aperti.

Il limite fondamentale di questi metodi è duplice. Da un lato richiedono che tutte le situazioni rilevanti siano anticipate in fase di progettazione, condizione non soddisfacibile in ambienti aperti e dinamici. Dall'altro, la transizione da simboli strutturati a compiti cognitivi ad alto livello come la percezione e il ragionamento in contesti aperti eccede sistematicamente le capacità degli approcci puramente logico-formali.

Un ulteriore ostacolo emerge con i concetti *fuzzy* relativi alle relazioni sociali e culturali degli esseri umani, che spesso difettano di interpretazioni consistenti tra contesti diversi. In questi scenari, le tecniche simboliche classiche si rivelano incapaci di elaborare adeguatamente le informazioni cognitive poiché le regole di alto livello non riescono a catturare le sfumature contestuali.

Per superare queste limitazioni, i modelli grafici probabilistici e i grafi della conoscenza sono diventati alternative più praticabili, grazie alla loro flessibilità, maggiore potere rappresentativo e scalabilità. Le tecniche neuro-simboliche più recenti impiegano entità primitive come prior di rappresentazione ma sfruttano reti neurali artificiali per apprendere le relazioni e le proprietà dei simboli.

13.5 Radicamento tramite Grafi della Conoscenza

Una delle strategie più mature e ampiamente impiegate per catturare relazioni espressive tra entità nei modelli linguistici è l'uso dei *Knowledge Graphs* (KG). I grafi della conoscenza rappresentano le parole come nodi in una struttura visiva arborescente nota come grafo o rete semantica. Tali nodi rappresentano oggetti individuali, categorie di oggetti, eventi e concetti; le relazioni tra le varie parole sono descritte dagli archi che connettono i nodi.

Formalmente, un grafo della conoscenza è una struttura $G = \langle \mathcal{V}, \mathcal{E}, \mathcal{R} \rangle$ dove $\mathcal{V}$ è l'insieme dei nodi-entità, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{R} \times \mathcal{V}$ è l'insieme delle triple relazionali, e $\mathcal{R}$ è il vocabolario delle relazioni. Una tripla (h, r, t) — con h testa, r relazione, t coda — codifica esplicitamente il fatto che tra h e t sussiste la relazione r.

Diverse famiglie di funzioni di scoring sono state proposte per apprendere rappresentazioni vettoriali delle entità e delle relazioni a partire da tali triple. Il modello TransE, tra i più elementari, definisce:

$$f(h, r, t) = \|\text{Emb}(h) + \text{Emb}(r) - \text{Emb}(t)\|_2$$

dove $\text{Emb}: \mathcal{V} \cup \mathcal{R} \to \mathbb{R}^d$ è la funzione di embedding. Il training minimizza la perdita a margine:

$$\mathcal{L} = \sum_{(h,r,t) \in \mathcal{P}} \sum_{(h',r,t') \in \mathcal{N}} \max(0, \gamma + f(h,r,t) - f(h',r,t'))$$

dove $\mathcal{P}$ è l'insieme delle triple positive, $\mathcal{N}$ quello delle triple negative ottenute per corruzione, e γ è un margine di sicurezza. Varianti più espressive come RotatE modellano ogni relazione come una rotazione nello spazio vettoriale complesso:

$$\text{Emb}(t) = \text{Emb}(h) \circ \text{Emb}(r), \text{ con } |\text{Emb}(r)| = 1 \text{ per ogni dimensione } k$$

dove $\circ$ denota il prodotto elemento per elemento di Hadamard nello spazio $\mathbb{C}^d$.

Questo meccanismo consente ai KG di immagazzinare un grande volume di conoscenza esplicita radicata nel mondo reale. Per questa ragione, sono stati proposti come strumento per mitigare problemi comuni come l'allucinazione e per fornire ai LLM conoscenza fisicamente radicata, riducendo la necessità di enormi dataset di addestramento. La conoscenza strutturata nei KG codifica relazioni esplicite ed è perciò più semanticamente

significativa e adatta per compiti emergenti come il ragionamento e la pianificazione, a differenza delle pure architetture neurali.

Un'ulteriore complementarità emerge dalla possibilità di impiegare i LLM stessi per costruire o arricchire i KG: grazie alla vasta conoscenza del mondo di cui dispongono, i modelli linguistici possono suggerire nuovi nodi, relazioni e vincoli che poi arricchiscono il grafo, producendo un ciclo di reciproco potenziamento. Queste due classi di metodi — LLM e KG — possono essere integrate in modo che si potenzino vicendevolmente, con i KG che migliorano le prestazioni degli LLM e il modello linguistico che aiuta a estendere e raffinare i grafi.

13.6 Radicamento tramite Prompting Ontologico

Una seconda famiglia di approcci sfrutta le tecniche di *prompting* per guidare i LLM verso risposte più sfumate e contestualmente appropriate. L'idea fondamentale è quella dell'apprendimento nel contesto (*in-context learning*): fornendo istruzioni o esempi espliciti — coppie input-output — alla fase di inferenza, è possibile ancorare le predizioni del modello a una struttura referenziale specifica senza alterare i parametri appresi.

Più recentemente, invece di fornire direttamente istruzioni leggibili dagli umani, numerosi lavori hanno proposto di sfruttare le *ontologie* come basi di conoscenza simbolicamente radicate in grado di produrre prompt contestualmente pertinenti in modo automatico. Un'ontologia è creata mediante una specifica formale di fatti, regole ed entità, categorie, proprietà e relazioni tra di esse. Diversi tipi di operazioni permettono di generare nuova conoscenza a partire da fatti e regole stabiliti in contesti specifici.

Formalmente, sia $\Omega = \langle \mathscr{C}, \mathscr{I}, \mathscr{R}_O, \mathscr{A}_O \rangle$ un'ontologia dove $\mathscr{C}$ è l'insieme dei concetti, $\mathscr{I}$ quello degli individui, $\mathscr{R}$_O le relazioni gerarchiche e compositive, e $\mathscr{A}$_O gli assiomi logici che vincolano le istanze. Un motore ontologico di prompting trasforma una query utente q in un insieme di prompt strutturati:

$$P(q \mid \Omega) = \{p_1(q, \omega_1), p_2(q, \omega_2), \ldots, p\ (q, \omega\ )\}$$

dove ciascun p_i è costruito a partire da un sotto-grafo $\omega_i \subseteq \Omega$ rilevante per q. Il LLM condizionato su $P(q \mid \Omega)$ produce predizioni che rispettano i vincoli ontologici, senza però richiedere un nuovo addestramento.

Il vantaggio principale è la possibilità di affinare rappresentazioni già radicate; il limite è la non scalabilità: non si può ottenere un radicamento esaustivo attraverso il prompting, il quale presuppone conoscenza dei concetti e delle relazioni sottostanti ed è quindi condizionato alla qualità e alla completezza dell'ontologia di partenza.

13.7 Radicamento End-to-End tramite Embedding Vettoriale

Un terzo approccio risolve il problema del radicamento modellandolo implicitamente nello spazio vettoriale ad alta dimensione. Nei LLM, i simboli come le parole e i concetti visivi sono codificati in base ai contesti in cui compaiono frequentemente e a come si relazionano

con altri simboli; i ricercatori hanno ideato tecniche che sfruttano questa rappresentazione per associare gli embedding appresi agli oggetti, alle esperienze percettive, alle azioni o ai concetti nel mondo reale.

Sia $\mathbb{S} = \{s_1, \ldots, s_{|\mathbb{V}|}\}$ il vocabolario e sia $E: \mathbb{S} \to \mathbb{R}^d$ la funzione di embedding appressa. La similarità semantica tra due simboli è misurata tipicamente dalla coseno-similarità:

$$\cos(s_i, s) = (E(s_i)^T E(s)) / (\|E(s_i)\|_2 \|E(s)\|_2)$$

La proprietà di composizionalità distributiva si manifesta nella ben nota relazione vettoriale:

$$E(\text{re}) - E(\text{uomo}) + E(\text{donna}) \approx E(\text{regina})$$

che suggerisce che le relazioni semantiche siano codificate come trasformazioni geometriche nello spazio di rappresentazione. In senso più generale, un sistema di radicamento end-to-end apprende una mappa:

$$G_\theta: \mathbb{R}^d \to \mathcal{W}$$

attraverso l'ottimizzazione di una funzione di perdita che incorpora vincoli sul mondo reale — siano essi costruiti da dati percettivi, motori o linguistici. Nell'approccio multimodale, per esempio, si minimizza una perdita contrastiva che rende vicine nello spazio latente le rappresentazioni di uno stesso concetto in modalità diverse (testo e immagine):

$$\mathcal{L}_{\text{contrastive}} = -\log(\exp(\text{sim}(E_T(s_i), E_I(v_i)) / \tau) / \sum_j \exp(\text{sim}(E_T(s_i), E_I(v)) / \tau))$$

dove E_T e E_I sono gli encoder testuali e visivi rispettivamente, τ è una temperatura di calibrazione, e (s_i, v_i) è una coppia testo-immagine allineata semanticamente.

Questa classe di tecniche è più scalabile e relativamente semplice da implementare rispetto agli altri metodi di radicamento discussi. Tuttavia è importante notare che tali approcci possono non produrre mappature precise in alcuni casi. Risulta anche difficile determinare quali simboli non siano correttamente radicati, e i metodi di radicamento che sfruttano gli embedding vettoriali tipicamente soffrono di problemi di affidabilità e mancanza di spiegabilità.

13.8 Radicamento per Esplorazione Attiva

Una quarta strategia, concettualmente la più ambiziosa, radica i simboli attraverso l'interazione diretta dell'agente con l'ambiente fisico o simulato. La logica sottostante è quella dell'*embodied cognition*: la conoscenza non si apprende passivamente da testi, ma è costruita attraverso cicli di azione-percezione-feedback che anchorano progressivamente i simboli astratti a esperienze sensorimotorie concrete.

Siano $a \in \mathcal{A}$ le azioni disponibili all'agente, $o \in \mathcal{O}$ le osservazioni percepite, e $r \in \mathbb{R}$ i segnali di rinforzo ricevuti dall'ambiente. Il processo di radicamento per esplorazione si formalizza come un problema di apprendimento per rinforzo in cui la funzione di ricompensa incorpora obiettivi semantici:

$r_t = r_estr(o , a) + \alpha \cdot r_sem(s , c)$

dove r_estr è la ricompensa estrinseca dall'ambiente, r_sem è una ricompensa semantica che premia la coerenza tra il simbolo s attivato al tempo t e il suo contesto referenziale c , e α è un coefficiente di bilanciamento.

L'agente apprende una policy ottimale:

$\pi^* = \text{argmax}_\pi \mathbb{E}_\pi [\sum_\{t=0\}^\{T\} \gamma^t r]$

dove $\gamma \in (0,1)$ è il fattore di sconto. Attraverso l'ottimizzazione di questa policy, l'agente impara progressivamente a connettere costrutti linguistici, oggetti fisici, concetti astratti e azioni.

Questa metodologia assicura un accoppiamento diretto tra informazione cognitiva e mondo reale, con rappresentazioni fisicamente plausibili. Il radicamento attivo è però costoso e dispendioso in termini di tempo da implementare, e può produrre connessioni incomplete se usato da solo. Non può nemmeno essere applicato a concetti puramente astratti che non abbiano un correlato fisico diretto.

Tra le realizzazioni concrete di maggior rilievo, EmbodiedGPT utilizza un'architettura a quattro componenti integrati: un vision-transformer per codificare le informazioni visive dalle osservazioni ambientali; un sottomodulo denominato Embodied-Former per mappare istruzioni testuali e informazioni visive e generare caratteristiche rilevanti per la pianificazione ad alto livello e il controllo di basso livello; un modello linguistico per i compiti linguistici; infine una rete di policy che genera azioni di basso livello dai tratti appresi dall'Embodied-Former. L'approccio *Chain of Thought* è utilizzato per generare sotto-obiettivi contestualmente pertinenti dai prompt.

13.9 Radicamento tramite Conoscenza Esterna

Una quinta famiglia di approcci sfrutta la conoscenza esterna da fonti diverse per fornire un radicamento *debole* ma praticamente efficace. I *grafi della conoscenza enciclopedici*, per esempio, possono rappresentare un grande volume di conoscenza strutturata estratta da fonti diverse, incluse enciclopedie come Wikipedia e basi di dati relazionali. Pur non coinvolgendo necessariamente entità primitive astratte connesse da regole sintattiche o logiche, offrono comunque un ponte tra la conoscenza implicita nei LLM addestrati su dataset generici e il mondo reale.

La *Retrieval-Augmented Generation* (RAG) è la tecnica più diffusa in questa categoria. L'idea fondamentale è quella di sfruttare informazioni aggiuntive da fonti esterne per potenziare la conoscenza disponibile per il processo di radicamento. Il meccanismo si formalizza come segue:

Date una query q e una base di conoscenza esterna $\mathcal{K} = \{d_1, d_2, \ldots, dN\}$, un retriever R seleziona il sottoinsieme più pertinente:

$\mathcal{D}_q = R(q, \mathcal{K}) = \{d_i : \text{score}(q, d_i) \geq \tau\}$

dove score: $\mathcal{Q} \times \mathcal{K} \to \mathbb{R}$ è una funzione di pertinenza (tipicamente la coseno-similarità tra embedding densi) e τ è una soglia. La predizione del LLM viene quindi condizionata sull'unione della query e dei documenti recuperati:

$\hat{y} = \text{LLM}(q \oplus \mathcal{D}_q)$

dove $\oplus$ denota la concatenazione. In questo schema, il radicamento non è integrato nei parametri del modello ma è realizzato dinamicamente a ogni inferenza attraverso il recupero di conoscenza contestuale.

Un limite rilevante di questo approccio è che le informazioni esterne possono non essere garantite come corrette, a causa di possibili errori sconosciuti, inconsistenze o informazioni incomplete. Possono anche verificarsi perdite complete di accesso alle informazioni esterne per vari motivi; e tali dati possono essere maggiormente esposti ad attori malevoli, mettendo a rischio la sicurezza del sistema.

13.10 Una Sintesi Critica e le Sfide Residue

La rassegna degli approcci al radicamento simbolico nei LLM rivela una tensione strutturale tra due esigenze opposte: la scalabilità — necessaria per operare su domini aperti e multilingui — e la fedeltà semantica — condizione per un ancoraggio genuino piuttosto che puramente statistico. Nessuna delle cinque famiglie di metodi descritte risolve soddisfacentemente questa tensione.

L'approccio tramite grafi della conoscenza sacrifica la scalabilità sull'altare della precisione: la costruzione manuale di KG completi è un compito non triviale, e la loro integrazione con i LLM introduce sfide ingegneristiche significative. L'approccio tramite embedding vettoriale è altamente scalabile ma opaco, incapace di distinguere tra associazioni statistiche spurie e radicamenti genuini. Il prompting ontologico è preciso e affidabile nella misura in cui l'ontologia è completa e aggiornata, condizione raramente soddisfatta in domini dinamici. Il radicamento attivo è il più promettente dal punto di vista teorico, ma rimane praticabile solo in ambienti simulati con interfacce sensorimotorie predefinite. Il RAG offre flessibilità ma introduce dipendenze esterne fragili.

La sfida fondamentale che accomuna tutti questi approcci è ciò che potremmo denominare il *problema del salto percettivo*: la connessione tra le rappresentazioni latenti di un LLM e le entità del mondo fisico non è mai diretta, ma sempre mediata da artefatti — testi, grafi, vettori, documenti — che sono essi stessi già simbolici. Il radicamento genuino, come quello che emerge nell'ontogenesi umana attraverso anni di interazione sensorimotoria con il mondo, richiede qualcosa che i sistemi puramente linguistici strutturalmente non possono fornire: l'esperienza in prima persona.

13.11 Approcci ibridi neurali-simbolici di terza generazione per il grounding

Il dibattito tra approcci simbolici e sub-simbolici al grounding ha prodotto, nell'ultimo decennio, una convergenza verso framework ibridi di terza generazione che combinano la scalabilità dell'apprendimento profondo con la precisione del ragionamento simbolico.

Questi framework superano sia le limitazioni degli approcci puramente simbolici (rigidità, mancanza di robustezza al rumore) sia quelle degli approcci puramente neurali (opacità, fragilità a variazioni distribuzionali, allucinazioni). La comprensione dei progressi più recenti in questo spazio è essenziale per valutare come il framework JEPA possa contribuire al problema del grounding.

L'architettura Neurosymbolic Concept Learner (NS-CL, Mao et al. 2019) rappresenta uno dei paradigmi di integrazione neuro-simbolica più influenti: le percezioni visive vengono processate da reti neurali per estrarre attributi di oggetti (colore, forma, materiale, dimensione), che vengono poi manipolati attraverso un programma simbolico di ragionamento per rispondere a domande visive in linguaggio naturale. La chiave di NS-CL è la separazione tra percezione (neurale, end-to-end differenziabile) e ragionamento (simbolico, discreto). Questa separazione permette al modello di generalizzare in modo composizionale: un sistema addestrato a rispondere a domande semplici ("di che colore è il cubo grande?") può rispondere a domande complesse che non ha mai visto durante l'addestramento ("ci sono più cubi rossi grandi o più sfere blu piccole?").

Nel contesto di JEPA e Var-JEPA, il meccanismo del prior condizionale appreso $p_\theta(s_y \mid s_x, z)$ svolge implicitamente una funzione di ragionamento: apprende a "dedurre" la rappresentazione del target dalla rappresentazione del contesto, catturando relazioni strutturali tra porzioni diverse dell'input. Per dati tabulari, questo equivale a imparare le dipendenze condizionali tra colonne: se la colonna "età" è nel contesto e la colonna "reddito" è nel target, il prior condizionale apprende la distribuzione di reddito condizionata all'età osservata. Questa struttura è analoga a un programma simbolico di inferenza bayesiana, ma implementata in modo differenziabile e appresa dai dati. L'estensione di Var-JEPA per incorporare esplicitamente strutture di ragionamento simbolico --- per esempio, constraint logici su quali colonne devono determinare quali altre --- è una direzione di ricerca che potrebbe migliorare significativamente la qualità del grounding.

13.12 Metriche formali per il grounding: definizioni e misure operative

La valutazione del grado di grounding di un sistema AI è notoriamente difficile: non esiste una metrica unica e universalmente accettata, e le misure disponibili tendono a catturare aspetti parziali del fenomeno. Formalizzare un insieme coerente di metriche operative per il grounding è un passo necessario sia per valutare i progressi dei framework JEPA in questa direzione sia per guidare lo sviluppo di architetture più groundate.

Una prima classe di metriche valuta il grounding attraverso la consistenza delle previsioni rispetto a interventi sul mondo. Un sistema è considerato parzialmente groundato se, quando si interviene su una proprietà fisica di un oggetto (ad esempio, modificando il colore di un cubo in una scena visiva), la rappresentazione latente dell'oggetto si aggiorna in modo coerente con l'intervento. Formalmente, definendo un operatore di intervento

do(x → x') che modifica una proprietà di x producendo x', il grounding è misurato come |Enc(x) − Enc(x')||x − x'| --- il rapporto tra la variazione della rappresentazione e la variazione dell'input. Un sistema completamente groundato ha questo rapporto costante rispetto alla proprietà fisica modificata; un sistema non groundato ha il rapporto arbitrariamente variabile.

Una seconda classe di metriche valuta il grounding attraverso la coerenza semantica delle analogie vettoriali. La classica relazione Enc("re") − Enc("uomo") + Enc("donna") ≈ Enc("regina") cattura un pattern di grounding attraverso la struttura geometrica dell'embedding space. Per Var-JEPA, una metrica analoga potrebbe essere: |Enc(Text_A) + Enc(Code_B) − Enc(Text_B) − Enc(Code_A)| dove (Text_A, Code_A) e (Text_B, Code_B) sono due coppie di viste con una relazione semantica definita (e.g., Code_A è la regex per "parole che iniziano per A" e Code_B è la regex per "parole che finiscono per B"). Un valore basso di questa norma indica che le relazioni semantiche tra le viste sono codificate in modo coerente e trasferibile nello spazio latente --- una forma di grounding strutturale debole ma misurabile. L'analisi t-SNE di Huang et al. (Figure 4 e 6) fornisce evidenza qualitativa di questa struttura, ma una formalizzazione quantitativa attraverso metriche di analogia vettoriale rimane da sviluppare.

Questa conclusione non è pessimistica: suggerisce piuttosto che il radicamento simbolico nei LLM debba essere concepito come un continuum, non come una condizione binaria. I sistemi attuali dispongono di un radicamento parziale, distribuito, approssimato — sufficiente per molti compiti pratici, ma sistematicamente insufficiente per la generalizzazione robusta in contesti nuovi e lontani dalla distribuzione di addestramento. Il passo successivo verso l'AGI richiederà architetture che integrino, in maniera organica, i canali simbolici con quelli percettivo-motori, superando la separazione modulare che ancora caratterizza i migliori sistemi disponibili.

19.14 Tassonomia strutturata dei limiti residui: quali sono fondamentali e quali contingenti

Una distinzione critica nell'analisi dei limiti residui dei tre framework è quella tra limitazioni fondamentali --- dovute a proprietà strutturali del paradigma computazionale che non possono essere superate senza cambiamenti architetturali radicali --- e limitazioni contingenti --- dovute a scelte specifiche di implementazione, scala o dataset che possono essere superate con risorse aggiuntive o migliori algoritmi. Confondere le due categorie porta a disperdere risorse di ricerca su problemi insolubili nel paradigma attuale, o viceversa, ad abbandonare prematuramente direzioni che potrebbero produrre risultati con i giusti investimenti.

I limiti fondamentali, nel senso delle barriere che nessun aumento di scale, dati o parametri può rimuovere, sono tre. Il primo è il problema del radicamento simbolico nel senso di Harnad: le rappresentazioni latenti di sistemi JEPA sono determinate dalla struttura statistica delle coppie di dati, e qualunque funzione di mapping tra simboli e referenti soddisfi questa struttura statistica è arbitrariamente inter-scambiabile --- il teorema di Putnam sulla non-unicità del radicamento. Il secondo è l'impossibilità della

normatività per sola correlazione statistica: il predittore di LLM-JEPA apprende a mappare Enc(Text) verso Enc(Code) in modo statisticamente consistente, ma questa consistenza non implica la comprensione della regola che governa la relazione testo-codice --- il problema di Kripke sulla regola-following. Il terzo è la completezza della semantica formale: qualunque sistema formale sufficientemente espressivo da modellare l'aritmetica contiene enunciati veri che non può dimostrare (Gödel) e non può definire la propria nozione di verità (Tarski). Questi tre limiti non sono specifici di JEPA, ma si applicano a qualunque sistema computazionale basato su apprendimento statistico.

I limiti contingenti, viceversa, sono superabili con approcci appropriati. La costruzione automatica di viste linguistiche per LLM-JEPA su corpus generici è un problema difficile ma non fondamentalmente impossibile --- richiede un meccanismo di augmentation semantica del linguaggio che non esiste ancora nella forma necessaria, ma che potrebbe emergere attraverso la combinazione di back-translation, paraphrase generation e constraint-based rewriting. La scalabilità di Var-JEPA alla visione e al video è una sfida di efficienza computazionale (il costo $O((D_feat)^2)$ del target posterior) che può essere affrontata con attenzione lineare, masking agressivo e dropout del posterior. La generalizzazione cross-domain di LLM-JEPA su task linguistici diversi dalla traduzione testo-codice richiede dataset con struttura di viste appropriate --- un problema di dataset construction che è risolvibile con investimento di risorse.

19.15 Quanto siamo vicini ai limiti fondamentali? Una valutazione quantitativa

Una questione importante per la pianificazione della ricerca è la seguente: data la distanza attuale tra i sistemi JEPA e i limiti fondamentali identificati, quanto progresso è ancora possibile prima di incontrare le barriere strutturali? Una risposta qualitativa è inevitabilmente speculativa, ma un'analisi quantitativa parziale è possibile attraverso metriche proxy.

Per il radicamento simbolico, una metrica proxy è la robustezza delle rappresentazioni a perturbazioni semanticamente equivalenti: due frasi con lo stesso significato dovrebbero produrre rappresentazioni vicine nello spazio latente, mentre due frasi con significati diversi dovrebbero produrre rappresentazioni lontane. I benchmark di robustezza semantica --- come Winogrande (Sakagami et al. 2021) per il ragionamento di buon senso e HANS (McCoy et al. 2019) per la comprensione del linguaggio --- mostrano che i LLM attuali, inclusi quelli addestrati con LLM-JEPA, falliscono sistematicamente su trasformazioni che non alterano il significato ma alterano la forma superficiale. Il gap rispetto al "ceiling" teorico del grounding perfetto è ancora molto ampio.

Per la causalità, una metrica proxy è la performance sui benchmark di causality understanding come CausalBench (Götz et al. 2024) e CAUSE (Chen et al. 2024), che valutano la capacità di inferire relazioni causali da dati osservazionali. I migliori LLM attuali raggiungono circa il 60-70% di accuratezza su questi benchmark, con un gap significativo rispetto al ceiling del 95%+ ottenibile con modelli causali strutturali espliciti su dataset con struttura causale nota. LLM-JEPA non è stato valutato su questi

benchmark, ma la sua architettura (basata su correlazioni cross-modale, non su strutture causali esplicite) non suggerisce miglioramenti sostanziali rispetto ai baseline LLM puri. Questa analisi conferma che la causalità rimane un limite fondamentale non affrontato dai framework JEPA attuali.

Capitolo 14 — L'Embodiment Computazionale: LLM Incarnati e Modelli Motori

14.1 Oltre la Mente Disincarnata

La filosofia della mente del Novecento ha progressivamente eroso la credibilità di una certa immagine cartesiana del pensiero: quella di una mente che elabora simboli in isolamento dal corpo e dal contesto fisico in cui quel corpo è immerso. Le moderne concettualizzazioni della cognizione biologica suggeriscono che i processi cognitivi nel sistema nervoso umano siano profondamente radicati nelle interazioni della mente con il corpo e con l'ambiente esterno. In questa visione, cervello, corpo e ambiente formano un sistema unificato in cui si influenzano e si co-determinano reciprocamente nell'orientare il comportamento intelligente.

Il concetto di plasticità neurale — tra i fenomeni cognitivi più importanti che potenziano il comportamento adattativo dell'intelligenza — presuppone anch'esso tale interazione mente-corpo-ambiente. È stato argomentato che per i sistemi di IA, al fine di risultare veramente intelligenti, questi — analogamente ai sistemi biologici — debbano necessariamente poter interagire con il mondo in modo fisico e ricevere feedback, imparando dai risultati di quelle interazioni fisiche. Secondo questa ipotesi, l'intelligenza artificiale può conseguire capacità cognitive comparabili all'intelligenza di livello umano solo se tale intelligenza fosse creata in un corpo fisico dotato della capacità di compiere azioni fisiche sull'ambiente.

I sistemi di IA incorporati (*embodied AI systems*) sono sistemi che — a differenza degli approcci tradizionali puramente digitali — hanno una manifestazione fisica tangibile attraverso cui possono percepire ed elaborare informazioni sensoriali, e interagire con il proprio ambiente. Mumuni e Mumuni (2025) sviluppano una trattazione sistematica dell'embodiment come fondamento dell'intelligenza artificiale generale, identificando quattro dimensioni analitiche fondamentali: goal-awareness, self-awareness, situational-awareness e deliberate action.

14.2 Il Fondamento dell'Intelligenza Goal-Directed

L'embodiment fornisce la struttura per il comportamento intrinsecamente orientato agli obiettivi. Un sistema di intelligenza artificiale incarnato possiede necessariamente *agency*, ovvero intraprende azioni intenzionali — azioni che desidera compiere e su cui ha pieno controllo. Questo comportamento orientato agli obiettivi è un requisito fondamentale per l'autonomia. Oltre a ciò, sistemi di IA dotati di ricche risorse sensorimotorie con possibilità illimitate di esplorare e interagire con l'ambiente raggiungeranno capacità intellettuali estese.

Un tale sistema intelligente possiederà necessariamente una percezione accurata e robusta del mondo e del proprio stato; dovrà poter agire sul mondo e influenzarlo in modo finalizzato; dovrà essere capace di adattarsi fluidamente alla complessa dinamica del mondo reale. Mentre gli approcci convenzionali all'intelligenza producono modelli intrinsecamente rigidi e meccanicistici, l'intelligenza incorporata è più flessibile e sfumata, capace di connettere esperienze oggettive con valori soggettivi, norme culturali e aspettative.

Il comportamento goal-directed risulta particolarmente importante nei compiti a lungo orizzonte, situazioni che implicano rinforzi ritardati (*delayed reward*), dove le azioni immediate non hanno una correlazione diretta con lo stato sensoriale corrente dell'agente. In tali scenari, l'agente intelligente seleziona le azioni sulla base di obiettivi intrinseci piuttosto che su istruzioni esplicite. Gli obiettivi nella cognizione biologica possono assumere forme diverse e provenire da fonti diverse: obiettivi impliciti guidati da bisogni biologici, obiettivi persistenti o temporanei per un particolare compito, istruzioni fornite da altri esseri umani.

La formalizzazione matematica del comportamento goal-directed nei modelli di reinforcement learning riconduce la pianificazione a lungo termine all'ottimizzazione di un'aspettativa di ritorno scontato. Sia S lo spazio degli stati, A quello delle azioni, P(s' | s, a) la funzione di transizione, R(s, a) la funzione di ricompensa, e $\gamma \in (0,1)$ il fattore di sconto. La policy ottimale è:

$\pi^*(s) = \text{argmax_a } Q^*(s, a)$

dove la funzione valore-azione ottimale Q*(s, a) soddisfa l'equazione di Bellman:

$Q^*(s, a) = R(s, a) + \gamma \sum\{s'\} P(s' \mid s, a) \max\{a'\} Q^*(s', a')$

Nei sistemi LLM-based, questa struttura formale viene realizzata in modo ibrido: il modello linguistico fornisce le capacità di pianificazione simbolica ad alto livello, mentre moduli specializzati per il controllo sensorimotorio gestiscono l'esecuzione delle azioni primitive.

14.3 Goal-Awareness nei Modelli Linguistici: Stato dell'Arte e Limiti

I LLM generici hanno dimostrato di esibire comportamenti orientati agli obiettivi in misura crescente. Ciononostante, la goal-awareness nei LLM fuori dalla scatola rimane limitata: valutazioni sperimentali mostrano una scarsa consapevolezza degli obiettivi, in particolare negli orizzonti a lungo termine.

Gli approcci per introdurre la goal-awareness nei LLM in maniera più intrinseca implicano la formulazione specifica degli obiettivi nel framework dell'apprendimento profondo. Sono state esplorate diverse strategie: l'incorporazione degli obiettivi di alto livello come prompt di input che guidano il modello; la messa a punto (*fine-tuning*) su dataset specifici curati con l'obiettivo desiderato in mente; l'uso di grafi della conoscenza per codificare informazioni sugli obiettivi e progettare moduli di pianificazione degli obiettivi; il reinforcement learning con feedback umano (RLHF).

In EmbodiedGPT, l'approccio Chain of Thought viene utilizzato per generare sotto-obiettivi per azioni incorporate: dato un obiettivo di alto livello G espresso in linguaggio naturale, il modello genera una sequenza di sotto-piani:

$G \rightarrow \{g_1, g_2, \ldots, g\}$

dove ogni g_i è un sotto-obiettivo più granulare che può essere eseguito attraverso un'azione primitiva o una sequenza breve di azioni. Sistemi avanzati possono sfruttare la goal-awareness intrinseca ad alto livello per abilitare agenti intelligenti a formulare autonomamente obiettivi di basso livello e perseguire obiettivi specifici del compito senza supervisione umana esplicita.

Un limite persistente è che la goal-awareness nei LLM rimane circoscritta a obiettivi di basso livello relativi a compiti specifici. Gli approcci attuali non possono abilitare obiettivi generali, onnicomprensivi o globali capaci di orientare il comportamento in qualsiasi situazione — quella che Mumuni e Mumuni distinguono come differenza tra *mission-awareness* (orientamento su un compito specifico o insieme correlato di compiti) e *goal-awareness* in senso pieno.

14.4 Situational-Awareness: Percezione Incarnata

La consapevolezza situazionale delle IA incorporata (*situational-awareness* degli agenti incarnati) coinvolge due aspetti principali: la consapevolezza dell'ambiente e la consapevolezza degli altri agenti strategici.

La consapevolezza dell'ambiente, nel contesto della cognizione incarnata, implica non solo la comprensione dello stato attuale del mondo e dei processi in esso in corso, ma anche la comprensione di come l'ambiente cambierà nel prossimo e nel lontano futuro come risultato di vari fattori, inclusi soprattutto gli effetti delle azioni dell'agente e/o di altri agenti. Per gli esseri umani, la consapevolezza situazionale è il risultato di conoscenza acquisita attraverso l'apprendimento e l'esperienza, di istinti e conoscenza innata trasmessa geneticamente, nonché di informazioni fornite da altri esseri umani e sistemi intelligenti.

La consapevolezza degli altri agenti è particolarmente rilevante nei contesti multiagente complessi, dove il comportamento è influenzato non solo da oggetti statici e inanimati, ma anche dalle azioni intelligenti e finalistiche di altri agenti, che possono essere cooperativi o competitivi in un determinato momento. In tali ambienti, gli esiti comportamentali dipendono dagli obiettivi, dall'intelligenza e dalle competenze complessive degli altri agenti — proprietà che non possono essere direttamente osservate ma vengono inferite dalle azioni e dalle reazioni degli agenti nel corso dell'interazione.

Approcci come Hypothetical Minds (HM) osservano la storia delle azioni degli altri agenti e sfruttano tali informazioni per predirne le strategie, producendo una descrizione ad alto livello in linguaggio naturale che può poi essere utilizzata per affinare il proprio comportamento. Questa struttura può essere formalizzata come un processo di inferenza bayesiana:

$P(\text{strat}_a \mid \text{hist}) \propto P(\text{hist} \mid \text{strat}_a) \cdot P(\text{strat}_a)$

dove $strat_a$ è la strategia dell'agente a, hist è la storia delle azioni osservate, $P(hist \mid strat_a)$ è la verosimiglianza delle azioni osservate dato che l'agente segue $strat_a$, e $P(strat_a)$ è la distribuzione a priori sulle strategie possibili.

14.5 Approcci per la Realizzazione dell'Embodiment

Mumuni e Mumuni identificano quattro principali classi di approcci per la realizzazione pratica dell'embodiment nei sistemi LLM.

Agenti fisici in ambienti reali. L'approccio più diretto consiste nel progettare e implementare agenti incorporati nella forma di robot con modalità sensoriali appropriate, integrando poi le avanzate capacità di comprensione linguistica con i meccanismi fisici e sensoriali del robot. Gli agenti generalisti incarnati devono svolgere simultaneamente molteplici compiti: percezione, pianificazione, navigazione, manipolazione di oggetti, comunicazione in linguaggio naturale, interazione fisica con esseri umani e altri agenti IA, nonché compiti di controllo a basso livello. I dataset più rappresentativi per questo tipo di addestramento — come EgoExoLearn, Holoassist, EgoTracks — consistono tipicamente di dataset egocentri contenenti video di esseri umani che svolgono azioni diverse in contesti differenti, con allineamento tra azioni e descrizioni linguistiche contestualmente pertinenti.

Agenti simulati in ambienti virtuali. Un approccio alternativo e computazionalmente più accessibile consiste nel creare e addestrare agenti virtuali in ambienti digitali 3D simulati. Questo offre un mezzo a basso rischio, rapido ed economico per apprendere informazioni sul mondo. In un ambiente simulato, gli agenti possono apprendere dagli esseri umani attraverso interfacce di interazione uomo-computer; possono anche apprendere dalle esperienze di altri agenti attraverso l'osservazione o l'interazione con loro. Questa osservazione condivisa potenzia esponenzialmente le capacità degli agenti intelligenti.

Ambienti in realtà estesa (XR). Le tecnologie di realtà estesa — in particolare la realtà virtuale e mista — possono creare ambienti 3D immersivi e altamente interattivi che simulano accuratamente il comportamento del mondo reale. In ambienti XR, gli agenti LLM incarnati possono sfruttare il feedback sensorimotorio simulato per imparare a percepire e agire nel mondo. In modalità mista, gli agenti virtuali *vivono* nel mondo reale e possono interagire fluidamente con esso nonché con altri oggetti virtuali. La piattaforma Social XR crea ambienti che consentono a esseri umani diversi di impegnarsi in attività pratiche centrate sull'uomo con oggetti virtuali, simulando interazioni uomo-agente e agente-agente.

Ambienti generati da IA. A causa della complessità degli ambienti virtuali simulati e degli elevati costi di manodopera nella loro creazione, un gran numero di approcci recenti ha proposto di sfruttare i modelli fondazionali preaddestrati come simulatori del mondo. Con questo approccio, i framework LLM sono appositamente costruiti per sfruttare la loro ricca conoscenza per generare ambienti di addestramento incorporati che fungano da media in cui addestrare altri agenti LLM e VLM.

14.6 Self-Awareness nei Sistemi Incarnati

La self-awareness è la capacità del sistema di IA di comprendere la propria natura, incluse le proprietà, le capacità, i limiti, il contesto e il ruolo nelle interazioni con entità esterne. La

struttura fisica del corpo influenza il modo in cui i sistemi biologici elaborano le informazioni: forma, dimensioni e capacità vincolano e consentono certi tipi di capacità e azioni, che a loro volta influenzano le strategie cognitive necessarie per realizzarle. I sistemi nervosi degli organismi viventi imparano naturalmente a controllare i meccanismi corporei — muscoli, arti — in modo specifico al corpo; questo spiega, per esempio, perché gli esseri umani richiedano un addestramento estensivo per essere in grado di usare agevolmente arti prostetici.

Un agente incarnato self-aware è anche capace di comprendere le implicazioni delle proprie azioni sugli altri agenti e sull'ambiente nel suo complesso. Un agente IA che sia sia self-aware che situational-aware è definito *situated*.

Le ricerche empiriche mostrano che i LLM multimodali all'avanguardia possiedono in generale la self-awareness come capacità emergente, una capacità che sorge spontaneamente come risultato del puro volume di dati di addestramento. In modo specifico, i modelli mostrano un certo grado di self-awareness riguardo ai limiti della propria conoscenza, ovvero la consapevolezza di ciò che non sanno. Diverse tecniche sono state proposte per elicitare ulteriori gradi di self-awareness: l'in-context learning, il reinforcement learning con intervento umano in the loop, e il fine-tuning su dataset curati per la self-awareness.

Un approccio recente consiste nel formulare il compito di self-awareness come una domanda di ricerca intuitiva, mediante la quale un agente incarnato interroga il proprio LLM di base sulla conoscenza del mondo esistente riguardo alla situazione data. Approcci più recenti hanno proposto di inferire le proprietà dei modelli LLM dalle rappresentazioni nascoste dei loro stati interni: SEAKR computa una cosiddetta *self-aware uncertainty* dalle rappresentazioni latenti degli stati interni della rete feedforward dell'LLM, confrontando il punteggio di consistenza attraverso risposte multiple.

La self-awareness può essere formalizzata come un problema di stima dell'incertezza epistemica. Sia f_θ il modello con parametri θ e sia x un input. L'incertezza epistemica U(x) può essere quantificata, in un framework bayesiano approssimato, come:

$$U(x) = \mathbb{E}_{\theta \sim q(\theta)}[f_\theta(x)^2] - (\mathbb{E}_{\theta \sim q(\theta)}[f_\theta(x)])^2$$

dove q(θ) è una distribuzione approssimata sui parametri. In pratica, tecniche come il MC-Dropout approssimano questa quantità attraverso campionamenti multipli con dropout attivo durante l'inferenza:

$$U(x) \approx (1/T) \sum_{t=1}^{T} f_{\theta,t}(x)^2 - ((1/T) \sum_{t=1}^{T} f_{\theta,t}(x))^2$$

dove $f_{\theta,t}$ è la predizione all'iterazione t con maschere di dropout diverse. Un LLM self-aware può riconoscere quando la propria conoscenza interna è insufficiente ad affrontare un problema e ricorrere a risorse aggiuntive.

14.7 Azione Deliberata: Dalla Pianificazione all'Esecuzione

Sebbene i LLM siano primariamente entità linguistiche, quando dotati di embodiment — sia come robot fisici, agenti virtuali o altri sistemi interattivi — possono intraprendere azioni deliberate nel mondo reale o attraverso interazioni virtuali o simulate. Questa capacità deriva dalla loro abilità di comprendere dialoghi orientati al compito o all'obiettivo, formulare piani passo-passo per realizzare compiti o raggiungere obiettivi target, ed eseguire azioni appropriate al compito in accordo con i piani predefiniti.

Un agente incarnato può scoprire nuove opportunità di azione (*affordances*) e proprietà precedentemente sconosciute degli oggetti attraverso l'interazione deliberata con l'ambiente. Questo gli consente di prendere decisioni o intraprendere azioni più benefiche, empatiche e moralmente informate.

I modelli Vision-Language-Action (VLA) costituiscono una nuova famiglia di modelli fondazionali multimodali specificamente progettati per eseguire azioni. Apprendono congiuntamente le modalità visive, linguistiche e di azione attraverso l'addestramento end-to-end. Di conseguenza, possono percepire l'ambiente, interpretare le istruzioni, effettuare pianificazione di alto livello e sintetizzare azioni di basso livello per completare vari compiti.

La struttura formale della pianificazione in questi sistemi può essere descritta come un processo di inferenza in un Markov Decision Process (MDP) gerarchico. A livello alto, un *task-level planner* genera una sequenza di sotto-compiti $\tau = \langle t_1, t_2, \ldots, t \rangle$ a partire da un obiettivo G e uno stato corrente s:

$$\tau^* = \text{argmax}_\{\tau\} \log P(\tau \mid G, s) = \text{argmax}_\{\tau\} \sum_i \log P(t_i \mid t_\{<i\}, G, s)$$

A livello basso, ciascun sotto-compito t_i viene tradotto in una sequenza di azioni primitive:

$$a_i^* = \text{argmax}_\{a\} P(a \mid t_i, o_i)$$

dove o_i è l'osservazione corrente dell'agente quando esegue t_i. I modelli VLA come RT-2, Unified-IO 2 e 3D-VLA realizzano questa struttura gerarchica combinando un backbone linguistico-visivo con un modulo di generazione delle azioni.

L'azione deliberata, secondo Mumuni e Mumuni, è limitata nei sistemi LLM attuali: la loro capacità di compiere azioni fisiche è circoscritta ad azioni semplici decomponibili in passaggi definiti. La realizzazione di comportamenti fisici complessi, continui e adattativi in ambienti aperti rimane una sfida aperta che richiederà architetture capaci di integrare la pianificazione simbolica con il controllo reattivo di basso livello in modo più organico di quanto i sistemi attuali non consentano.

14.8 Riflessioni Teoriche sull'Embodiment come Fondamento dell'AGI

Il quadro dell'embodiment delineato da Mumuni e Mumuni si rivela profondamente connesso con le questioni più generali che attraversano la riflessione sull'intelligenza artificiale generale. Tre nodi teorici meritano particolare attenzione.

Il primo riguarda la relazione tra embodiment e radicamento simbolico trattata nel capitolo precedente. L'embodiment non è semplicemente uno strumento per il radicamento: è, in senso più profondo, la condizione di possibilità dello stesso. Le esperienze incarnate servono come segnali utili per il radicamento dei simboli — ovvero, piuttosto che dipendere esclusivamente da associazioni linguistiche astratte dei termini di addestramento, l'embodiment consente all'agente di ottenere esperienze sensorimotorie significative, attraverso la percezione effettiva e le interazioni con il mondo, che vengono poi usate per radicare le rappresentazioni astratte. Le rappresentazioni incarnate radicate consentono poi all'agente di osservare e apprendere le relazioni causali direttamente dal mondo fisico attraverso l'interazione con esso.

Il secondo nodo riguarda il *Sim-to-Real Gap*: il divario tra le competenze acquisite in ambienti simulati e le prestazioni effettive nel mondo fisico reale. Questo problema non è puramente tecnico ma riflette una questione epistemica più profonda: la simulazione, per quanto accurata, rimane un modello del mondo e non il mondo stesso. Le semplificazioni necessarie per rendere computazionalmente trattabile la simulazione introducono sistematicamente bias che si manifestano come degrado delle prestazioni al momento del trasferimento. Questo problema non ha una soluzione ovvia finché non si disponga di simulatori capaci di replicare con fedeltà arbitraria tutti gli aspetti fisicamente rilevanti dell'ambiente.

Il terzo nodo riguarda la distinzione tra autonomia *missionale* e autonomia *globale*. Nella cognizione biologica si possono gestire obiettivi generali di alto livello, mentre i LLM, fino ad oggi, sono generalmente limitati a compiti che possono essere descritti o decomposti in sotto-compiti multipli, ciascuno consistente in una sequenza di passaggi in cui esiste un certo tipo di ordinamento fisso delle azioni. Questa forma di goal-awareness può essere più accuratamente descritta come *mission-awareness*. Nei complessi scenari del mondo reale, gli agenti intelligenti hanno bisogno di comprendere non solo gli obiettivi immediati, ma anche come molteplici obiettivi intermedi — spesso apparentemente contraddittori — si inseriscano in contesti più ampi, compresi gli interessi della società nel suo complesso. Raggiungere questa forma superiore di goal-awareness rimane uno dei problemi centrali nell'agenda della ricerca verso l'AGI.

14.9 La formalizzazione matematica dell'embodiment: dalla percezione all'azione attraverso i modelli del mondo

L'embodiment, nella sua accezione più rigorosa, non è semplicemente la presenza di un corpo fisico ma la strutturazione della cognizione attraverso l'interazione continua tra corpo, cervello e ambiente. Formalizzare questo concetto in termini matematicamente precisi è essenziale per progettare architetture AI genuinamente incarnate e per valutare il contributo delle architetture JEPA a questa sfida.

Un framework formale influente è quello della Active Inference (Friston 2010), derivato dal principio di free energy minimization. In questo framework, un agente incarnato minimizza la sua sorpresa (variational free energy) rispetto alle sue osservazioni attraverso due meccanismi complementari: l'inferenza percettiva (aggiornare le credenze sullo stato del mondo per ridurre la discrepanza tra predizioni e osservazioni) e l'azione

(modificare l'ambiente per rendere le osservazioni più coerenti con le predizioni). Formalmente, la variational free energy è: F = E_q(s|o) [-log p(o|s)] + KL(q(s|o) || p(s)), dove o sono le osservazioni, s lo stato nascosto del mondo, p(o|s) il modello generativo del mondo, p(s) il prior sullo stato, e q(s|o) la distribuzione variazionale a posteriori. Il primo termine è l'errore di ricostruzione, il secondo la pressione verso il prior --- una struttura formalmente analoga all'ELBO di Var-JEPA.

La connessione tra Active Inference e Var-JEPA è strutturalmente profonda: entrambi i framework massimizzano un limite inferiore sulla log-evidenza del modello generativo. La differenza fondamentale è che in Active Inference il processo generativo è il world model dell'agente incarnato --- produce previsioni su osservazioni sensoriali attraverso il corpo e l'ambiente --- mentre in Var-JEPA il processo generativo opera su coppie di rappresentazioni latenti di dati statici. Estendere Var-JEPA verso Active Inference richiederebbe di: (1) aggiungere un modello di transizione p_θ(s_t | s_{t-1}, a_t) che descriva come lo stato latente evolve in funzione delle azioni, (2) aggiungere un modello di policy π_φ(a_t | s_t) che generi azioni a partire dallo stato attuale, e (3) ottimizzare congiuntamente il modello generativo (per ridurre la sorpresa) e la policy (per selezionare azioni che riducano la sorpresa futura). Questa architettura corrisponde esattamente a un World Model JEPA variazionale --- la visione di LeCun per il sistema AMI, realizzata attraverso il formalismo bayesiano di Var-JEPA.

14.10 World Models come ponte architetturale tra JEPA e agenti incarnati: DreamerV3 e oltre

La famiglia di World Models basati su RSSM (Recurrent State Space Model) di Hafner --- da DreamerV1 (2020) a DreamerV3 (2023) --- rappresenta l'istanza più sviluppata e empiricamente validata dell'idea di apprendimento delle rappresentazioni come fondamento dell'intelligenza incarnata. Comprendere le relazioni formali tra DreamerV3 e Var-JEPA è importante per valutare come i progressi teorici del paradigma variazionale possano essere applicati a sistemi di reinforcement learning basati su modelli del mondo.

DreamerV3 apprende un world model composto da: (1) un encoder h_t = f_φ(h_{t-1}, z_t, a_{t-1}) che aggiorna uno stato ricorrente h_t, (2) un modello di rappresentazione z_t ~ q_φ(z_t | h_{t-1}, x_t) che inferisce il latente corrente dall'osservazione, (3) un modello di transizione ẑ_t ~ p_θ(ẑ_t | h_t) che predice il latente senza l'osservazione (il "world model" in senso stretto), e (4) decoder per osservazioni, reward e terminazione. L'obiettivo di addestramento è un ELBO su sequenze: max_{θ,φ} Σ_t [E_q log p_θ(x_t | z_t) + βrec · E_q log p_θ(r_t | z_t) − βkl · KL(q_φ || p_θ) + ...]. La struttura è analoga all'ELBO di Var-JEPA: termini di ricostruzione (per x_t e r_t) e termine KL (tra posteriore variazionale e prior del world model).

La differenza fondamentale tra DreamerV3 e Var-JEPA non è nella struttura matematica ma nella domanda a cui i due sistemi rispondono. DreamerV3 risponde alla domanda: "come pianificare comportamenti efficaci in ambienti complessi attraverso l'immaginazione nel spazio latente?". Var-JEPA risponde alla domanda: "come apprendere rappresentazioni latenti di alta qualità da dati statici attraverso la predizione

in spazio latente?". La sintesi dei due framework --- un Var-JEPA che opera su sequenze temporali con azioni, nel contesto di un agente incarnato che esplora un ambiente fisico o simulato --- è la visione a lungo termine del programma JEPA→AMI→AGI di LeCun, e il framework variazionale di Gögl e Yau fornisce le fondamenta teoriche principiate su cui questa sintesi può essere costruita.

Capitolo 15 — La Causalità nei Modelli Linguistici di Grande Scala: Oltre la Correlazione Statistica

15.1 Il Problema della Causalità nell'Intelligenza Artificiale

Tra i quattro pilastri fondazionali che Mumuni e Mumuni (2025) identificano come condizioni necessarie per il conseguimento dell'intelligenza artificiale generale, la causalità occupa una posizione peculiare: mentre embodiment e radicamento simbolico riguardano il modo in cui un sistema acquisisce conoscenza del mondo attraverso l'esperienza, la causalità riguarda la struttura profonda di quella conoscenza stessa, ovvero la capacità di distinguere le relazioni che *producono* fenomeni da quelle che semplicemente li *accompagnano*. Questa distinzione — tra causa ed effetto da un lato, correlazione statistica dall'altro — è epistemicamente fondamentale e praticamente decisiva.

La causalità caratterizza il modo in cui vari fattori, fenomeni o eventi influenzano altri eventi, oggetti o processi nel mondo reale. L'apprendimento causale nella sua forma più elementare mira a determinare le relazioni dinamiche tra due variabili, dove una variabile — la causa — influenza direttamente un'altra — l'effetto. Mentre il radicamento si occupa principalmente di connettere simboli primitivi come le parole alle loro rappresentazioni significative nel mondo fisico, la causalità si preoccupa di spiegare i meccanismi e le ragioni sottostanti ai cambiamenti di questi parametri e di come influenzino vari esiti nel mondo.

Il ragionamento causale permette ai sistemi di IA di fare previsioni accurate su fenomeni complessi del mondo reale — dalla resistenza delle strutture agli elementi atmosferici avversi al cambiamento climatico, dalla diffusione delle malattie alla crescita demografica, dalle prestazioni economiche agli incidenti. Tale comprensione è vitale per attività quotidiane come cucinare, lavare e guidare. Un ulteriore ruolo cruciale del ragionamento causale è migliorare la robustezza alle interferenze e mantenere inferenze corrette quando le condizioni sottostanti, i meccanismi interni o l'ambiente cambiano.

15.2 La Gerarchia Causale di Pearl

La struttura teorica più influente per la classificazione delle relazioni causali è quella formalizzata da Judea Pearl nel suo schema gerarchico a tre livelli. Pearl descrive un'architettura a tre gradini per classificare le relazioni causali derivate dalle osservazioni, o più precisamente dai dati. I livelli sono designati come Associazione (Livello 1), Intervento

(Livello 2) e Controfattuale (Livello 3); secondo questo schema, risolvere problemi di ragionamento a qualsiasi livello è possibile solo se sono disponibili informazioni da quel livello o da quelli superiori.

Livello 1 — Associazione. Il livello più basso del ragionamento causale riguarda situazioni in cui le risposte alle domande si ottengono direttamente dalle osservazioni sotto forma di relazioni statistiche nei dati osservati. La quantità fondamentale è la distribuzione condizionale:

$$P(Y \mid X = x)$$

che misura la probabilità dell'evento Y dato che X assuma il valore x. Questo livello risponde a domande del tipo "Cosa è X? Come osservare X modifica la nostra credenza su Y?" Il linguaggio matematico nativo è la teoria della probabilità bayesiana e le statistiche descrittive. Il limite epistemico è chiaro: una forte correlazione statistica $P(Y|X)$ non implica alcuna relazione causale tra X e Y, poiché entrambe le variabili potrebbero essere effetti di una causa comune latente.

Livello 2 — Intervento. Il secondo livello del ragionamento causale implica la stima della misura in cui cambiare una variabile (per esempio un'opzione di trattamento) influisce su una variabile obiettivo (ovvero un particolare esito). Pearl introduce l'operatore *do*, che formalizza l'intervento come mutilazione del grafo causale:

$$P(Y \mid do(X = x))$$

Questa distribuzione — l'*interventional distribution* — differisce radicalmente dalla distribuzione osservazionale $P(Y|X=x)$: mentre la seconda condiziona sulle osservazioni (lasciando intatta la struttura causale che genera X), la prima impone il valore di X per manipolazione esterna (recidendo tutti gli archi in entrata a X nel grafo). Il teorema del *do-calculus* di Pearl fornisce le regole algebriche per derivare distribuzioni interventive da dati osservazionali quando sono disponibili informazioni strutturali sufficienti. Le informazioni causali a questo livello consentono di predire correttamente l'effetto di azioni specifiche.

Livello 3 — Controfattuale. Il livello più alto consente di rispondere a domande ipotetiche o di fare inferenza su esiti non osservati. Questo comporta rispondere a domande del tipo "cosa sarebbe successo?" — ovvero cosa avrebbe potuto verificarsi se certi eventi non fossero accaduti o fossero accaduti diversamente. La quantità fondamentale è:

$$P(Y_{x'} = y \mid X = x, Y = y')$$

che si legge come "la probabilità che Y sarebbe stato y' se X fosse stato x', dato che in realtà abbiamo osservato X=x e Y=y". Questa quantità non è identificabile dalla distribuzione osservazionale o interventiva da sola: richiede informazioni sulla risposta individuale delle unità all'intervento, catturate attraverso i *potential outcomes* $Y_x(u)$ per ogni unità u.

Il ragionamento controfattuale consente di determinare quale variabile manipolare, e in quale misura, per portare una variabile obiettivo a uno stato desiderato. Risolvere problemi di intelligenza a questo livello implica usare sia informazioni associative che interventive.

15.3 Scoperta e Inferenza Causale

Apprendere la causalità (o costruire modelli causali) mira generalmente a risolvere due problemi: la *scoperta causale* (*causal discovery*), che identifica i meccanismi sottostanti, i parametri fisici associati e le interrelazioni che governano il funzionamento del sistema; e l'*inferenza causale* (*causal inference*), il compito di stimare gli effetti delle variabili causali l'una sull'altra sulla base di un'ipotesi predefinita sulle loro relazioni causali.

La scoperta causale da dati osservazionali è un problema di identificazione strutturale: dati n variabili $X_1, \ldots, X$ e T campioni da una distribuzione congiunta $P(X_1, \ldots, X)$, si cerca il DAG (Directed Acyclic Graph) $G = \langle V, E \rangle$ tale che la distribuzione sia Markoviana rispetto a G, ovvero fattorizzi secondo:

$$P(X_1, \ldots, X) = \prod_i P(X_i \mid Pa(X_i))$$

dove $Pa(X_i)$ denota l'insieme dei genitori di X_i nel DAG. La classe di equivalenza di Markov — i.e., l'insieme di tutti i DAG che codificano le stesse indipendenze condizionali — è identificata dal *CPDAG* (Completed Partially Directed Acyclic Graph), che può essere stimato da dati con algoritmi come PC e FCI o tramite metodi score-based come GES.

L'inferenza causale, dato G, procede attraverso la formula dell'aggiustamento. Per una struttura causale senza variabili confondenti non osservate, la distribuzione interventiva si calcola come:

$$P(Y \mid do(X = x)) = \sum_z P(Y \mid X = x, Z = z)\, P(Z = z)$$

dove Z è il set di aggiustamento che blocca tutti i *backdoor paths* da X a Y nel grafo. Il *backdoor criterion* stabilisce che Z è un set di aggiustamento valido se: (i) non contiene discendenti di X e (ii) blocca tutti i percorsi che entrano in X con un arco dal nodo precedente (percorsi backdoor).

15.4 Metodi di Deep Learning per la Causalità

I LLM multimodali addestrati su dati generici su larga scala hanno dimostrato una grande capacità di modellare relazioni causali. Questo è principalmente ottenuto apprendendo pattern nascosti dall'enorme quantità di dati di addestramento. I modelli possono essere in grado di inferire variabili importanti che influenzano la crescita economica o l'inflazione, nonché identificare specifici legami causali tra queste variabili attraverso pattern puramente appresi, anche quando tali informazioni non sono esplicitamente specificate nei dati di addestramento.

Tuttavia, la conoscenza acquisita in questo modo è vincolata dal fatto che non tutte le connessioni osservate hanno relazioni causa-effetto. Un gran numero di fenomeni del mondo reale mostrano relazioni correlative — una relazione in cui i cambiamenti nelle variabili obiettivo, per coincidenza o per influenze non correlate, si seguono l'uno all'altro senza che esista un legame causale reale. Nel ragionamento, tali false correlazioni possono condurre a conclusioni imprecise o errate.

In aggiunta, nonostante l'apparente potere di modellare la causalità, i ricercatori hanno dimostrato che i LLM all'avanguardia non sono in grado di acquisire reali capacità di ragionamento causale, anche con addestramento aggiuntivo mirato all'abilitazione della causalità — incluso in-context learning e fine-tuning. I LLM addestrati puramente sui dati, senza meccanismi interni di modellazione causale, non sono intrinsecamente consapevoli delle leggi fisiche o dei meccanismi e principi sottostanti che governano il comportamento nel mondo reale, e le loro previsioni si basano spesso su correlazioni apprese. Questo può portare a errori seri e pericolosi.

La distinzione formale tra i due regimi è netta. Sia f_θ un LLM addestrato su una distribuzione P_data che approssima P(Y|X). Il modello impara:

f_θ(X) ≈ 𝔼_{P_data}[Y | X]

Questa aspettativa condizionale è una quantità puramente osservazionale: cattura la correlazione statistica ma non l'effetto causale. Quando la distribuzione interventiva P(Y|do(X)) diverge da P(Y|X) — il che accade ogni volta che esistono confondenti non osservati — il modello produce previsioni sistematicamente errate.

15.5 Metodi Neuro-Simbolici per la Causalità

In contrasto con l'approccio di deep learning che costruisce modelli predittivi che apprendono principalmente dipendenze statistiche, i metodi neuro-simbolici incorporano esplicitamente conoscenza a priori sui meccanismi causali nel modello LLM. Un modo per raggiungere questo obiettivo è sfruttare i grafi della conoscenza e altre tecniche di rappresentazione e inferenza causale basate su conoscenza strutturata.

Uno degli approcci neuro-simbolici più efficaci per estendere la capacità di ragionamento causale dei LLM è integrare i *Modelli Causali Grafici* (SCM — Structural Causal Models), una classe speciale di tecniche di conoscenza strutturata che, per loro natura, sono intrinsecamente causali. L'approccio base consiste nel rappresentare formalmente le assunzioni causali usando diagrammi o grafi speciali.

Un SCM su n variabili è definito da un sistema di equazioni strutturali:

$X_i := f_i(Pa(X_i), U_i)$, per i = 1, …, n

dove f_i è la funzione causale per X_i, $Pa(X_i)$ sono i genitori di X_i nel DAG associato, e $U_i \sim P(U_i)$ sono rumori esogeni indipendenti. Questa struttura consente di calcolare:

— Distribuzioni osservazionali: $P(X_1, \ldots, X)$ mediante la fattorizzazione di Markov; — Distribuzioni interventive: $P(X \mid do(X_i = x))$ mediante la mutilazione dell'equazione $X_i := f_i(\ldots)$ sostituita da $X_i := x$; — Controfattuali: Y_{x}(u) := f_Y(Pa(Y) | do(X=x), U=u) mediante la risposta individuale al trattamento.

La proprietà cruciale degli SCM è che operano a tutti e tre i livelli della gerarchia di Pearl simultaneamente. La rappresentazione nella grafo causale permette di valutare la validità delle assunzioni causali: si può stabilire se e come variabile manipolare e in quale misura per

portare la variabile obiettivo a uno stato desiderato. I metodi con modelli causali grafici hanno dimostrato il potenziale di questo approccio.

Wang et al. propongono un sottomodulo chiamato Causal Relationship Enhancement (CRE) che utilizza Structural Causal Model per modellare i meccanismi causali per la successiva integrazione in un framework LLM. Samarajeewa et al. sfruttano conoscenza causale esterna per aumentare un LLM e migliorare il ragionamento causale, utilizzando la tecnica RAG per recuperare Grafi Causali come fonti di conoscenza esterna.

15.6 Modelli del Mondo Informati dalla Fisica

Una terza via per dotare i LLM di capacità causali genuine si fonda sull'ipotesi che la capacità umana di inferire e ragionare su eventi causali si basi su un *modello del mondo* o *modello mentale* che codifica astrazioni causali di concetti, fenomeni e oggetti in modo da mantenere una struttura e regole di comportamento definite, seppur sfumate. Sulla base di tali astrazioni, gli esseri umani hanno conoscenza approssimativa, implicita del mondo sotto forma di *fisica intuitiva* — ovvero proprietà elementari di varie entità e come queste proprietà influenzino il comportamento.

Numerosi lavori recenti sfruttano mondi virtuali basati su motori di fisica intuitiva per radicare la conoscenza dei LLM nelle proprietà causali e nel comportamento del mondo reale. Questi modelli impiegano modelli matematici formali progettati sulla base di conoscenza a priori per rappresentare le leggi fisiche sul mondo. Simulano relazioni causali ed effetti di fenomeni del mondo reale come aerodinamica, gravità, forza, luce e calore.

La forza di questo approccio risiede nella precisione e non ambiguità della conoscenza causale che codifica. Le relazioni matematiche sottostanti alle interazioni fisiche sono invariabilmente radicate nella fisica newtoniana rigorosa, nella termodinamica, o — a seconda del livello di realismo richiesto — nella fisica delle particelle e nella meccanica quantistica. A differenza dei modelli neurali puri, i simulatori fisici rispettano strutturalmente le seguenti proprietà invarianti:

Conservazione dell'energia: dE/dt = W_ext − Q_diss

dove W_ext è il lavoro delle forze esterne e Q_diss è il calore dissipato per attrito. Questa equazione costituisce un vincolo causale duro che nessuna approssimazione statistica può violare impunemente.

Conservazione della quantità di moto: d(mv)/dt = F_netta

Per forze conservative, l'equazione di Hamilton governa la dinamica:

$\dot{H} = \partial H/\partial t$, $\dot{p}_i = -\partial H/\partial q_i$, $\dot{q}_i = \partial H/\partial p_i$

dove H è l'Hamiltoniana, q_i le coordinate generalizzate e p_i i momenti coniugati. Un agente addestrato in un ambiente che rispetta questi vincoli apprende, almeno implicitamente, leggi causali generalizzabili sui movimenti, le cadute, le collisioni degli oggetti.

Il limite principale di questo approccio è che simulazioni altamente dettagliate su larga scala richiedono spesso una potenza di elaborazione significativa. È anche enormemente difficile modellare relazioni fisiche così dettagliate e precisamente. Le simulazioni potrebbero non riuscire a tenere conto accuratamente di concetti fuzzy o fattori sconosciuti agli sviluppatori umani.

15.7 La Tensione Fondamentale: Scalabilità versus Fedeltà Causale

La rassegna degli approcci alla modellazione causale nei LLM rivela una tensione strutturale analoga a quella osservata per il radicamento simbolico, ma con una specificità aggiuntiva: qui la tensione è tra la *completezza causale* — la capacità di rispondere a domande a tutti e tre i livelli della gerarchia di Pearl — e la *scalabilità computazionale* — la possibilità di applicare i metodi a domini aperti con strutture causali complesse e parzialmente osservabili.

Un confronto schematico delle capacità dei diversi approcci rivela il seguente quadro. I metodi di deep learning puro operano solo a livello 1 (associazione), non richiedono conoscenza a priori sul dominio ed è facile scalare, ma possono solo stabilire correlazioni, richiedono dataset specifici per la causalità e sono inclini a errori seri e catastrofici. I grafi della conoscenza operano a livello 1 e parzialmente a livello 2, possono modellare relazioni complesse e si integrano facilmente nei framework di deep learning, ma non tutti i tipi di dati sono supportati e la modellazione di relazioni causali può essere laboriosa e difficile da scalare. Gli SCM operano a tutti e tre i livelli, possono modellare le dipendenze causali in modo esaustivo, ma sono difficili da implementare, richiedono conoscenza completa di tutti i fattori e sono estremamente difficili da scalare. I modelli fisici operano anch'essi a tutti e tre i livelli e sono estremamente scalabili grazie ai modelli del mondo esistenti, ma richiedono enormi risorse computazionali e conoscenza dei fattori e delle loro relazioni.

Questa lettura comparativa suggerisce che nessun approccio singolo sia sufficiente. La direzione più promettente sembra essere quella di combinazioni ibride che sfruttino la scalabilità dei metodi neurali per la scoperta causale iniziale, la precisione degli SCM per la validazione delle ipotesi causali critiche, e la fisica informata per la generalizzazione in domini con strutture fisiche note. L'ulteriore sfida consiste nell'integrare queste componenti in un'architettura coerente capace di ragionare causalmente in modo fluido, adattivo e contestualmente sensibile — una sfida che rimane largamente aperta.

15.8 Il Ragionamento Causale come Condizione per l'AGI

La riflessione sul posto della causalità nell'intelligenza artificiale generale porta a una conclusione che il framework di Mumuni e Mumuni rende esplicita nella sua sezione di discussione: sistemi addestrati puramente sui dati, senza meccanismi interni di modellazione causale, tendono a produrre previsioni basate su correlazioni apprese, il che può condurre a errori seri e pericolosi.

Questo giudizio ha implicazioni dirette per la direzione della ricerca futura. Un sistema capace di ragionamento controfattuale — "cosa sarebbe successo se…?" — possiede una forma di comprensione del mondo qualitativamente diversa da quella di un sistema che interpola tra pattern osservati. La mente umana, nota Pearl, non è mai stata un semplice motore associativo; essa costruisce modelli causali del mondo che le consentono di valutare l'efficacia di interventi, imparare da controfattuali immaginari e pianificare azioni in scenari non ancora osservati.

Verso questa direzione, i lavori recenti che impiegano LLM per costruire grafi causali e poi utilizzarli per aumentare ulteriormente gli LLM stessi rappresentano un paradigma emergente di eccezionale interesse: il modello diventa sia lo strumento per la scoperta causale sia il beneficiario della conoscenza causale estratta. Questa struttura ricorsiva — in cui conoscenza causale e capacità di ragionamento si co-rafforzano iterativamente — potrebbe costituire uno dei percorsi più promettenti verso sistemi dotati di comprensione causale genuinamente robusta.

15.9 Intervento e controfattuale nello spazio latente JEPA: un programma di ricerca

La connessione tra JEPA e causalità è più profonda di quanto appaia superficialmente: il prior condizionale appreso $p_\theta(s_y \mid s_x, z)$ in Var-JEPA è formalmente un meccanismo causale nello spazio latente, nel senso che specifica come s_y "dipende da" s_x e z attraverso una funzione appressa. Tuttavia, come discusso nel Capitolo 19, questa dipendenza è osservazionale piuttosto che interventistica: il prior condizionale cattura la distribuzione di s_y dato s_x e z, non la distribuzione di s_y dopo un'intervento $do(s_x = \tilde{s}_x)$.

La strada per superare questa limitazione passa attraverso la formulazione di un Structural Causal Model (SCM) nello spazio latente di Var-JEPA. Formalmente, uno SCM latente per Var-JEPA assegnerebbe a ciascuna variabile latente una funzione strutturale: $s_x := f_x(U_x)$ (il contesto è generato da una causa esogena), $z := f_z(U_z)$ (variabile ausiliaria generata indipendentemente), $s_y := f_y(s_x, z, U_y)$ (il target è causato dal contesto, dalla variabile ausiliaria e dal rumore esogeno). La distribuzione interventistica $p(s_y \mid do(s_x = \tilde{s}_x))$ si calcola allora come: $p(s_y \mid do(s_x = \tilde{s}_x)) = \int f_y(\tilde{s}_x, z, U_y) \cdot p(U_y) \cdot p(z)\, dU_y\, dz$, che differisce dalla distribuzione osservazionale $p(s_y \mid s_x = \tilde{s}_x)$ quando esiste un backdoor path da s_x a s_y attraverso variabili confondenti latenti.

Un approccio sperimentale per approssimare il ragionamento interventistico in Var-JEPA è il seguente. Durante il fine-tuning o il test, si modifica artificialmente la rappresentazione del contesto $s_x \rightarrow s_x + \delta$ per una perturbazione δ specifica (ad esempio, una perturbazione nella direzione della feature semantica "lunghezza della sequenza") e si misura come cambia la distribuzione del prior condizionale $p_\theta(s_y \mid s_x + \delta, z)$. Se il prior condizionale risponde in modo coerente con l'intervento --- aumentando la probabilità di s_y corrispondente a sequenze target più lunghe quando si aumenta s_x nella direzione della lunghezza --- questo costituisce un'evidenza preliminare di sensibilità interventistica. Questo tipo di analisi di sensibilità, analoga alle

"concept activation vectors" (TCAV, Kim et al. 2018), è implementabile con strumenti esistenti e potrebbe fornire insight meccanistici sulle strutture causali implicite nel world model di Var-JEPA.

15.10 Apprendimento della struttura causale attraverso la predizione in spazio latente: connessioni con CausalVAE e DEAR

I recenti framework di Causal Representation Learning (CRL) --- tra cui CausalVAE (Yang et al. 2021), DEAR (Shen et al. 2022) e iVAE causale (Khemakhem et al. 2020) --- cercano di apprendere rappresentazioni latenti che corrispondono a fattori causali del processo generativo dei dati. Questi framework condividono con Var-JEPA la struttura di base (ELBO variazionale su modelli latenti) ma aggiungono vincoli causali espliciti che orientano le rappresentazioni verso fattori semanticamente interpretabili.

CausalVAE struttura il modello generativo come un DAG causale sui fattori latenti: $p(z_1, ..., z_d) = \Pi_i p(z_i \mid Pa(z_i))$, dove $Pa(z_i)$ sono i "genitori causali" di z_i nel grafo. Il grafo causale può essere specificato a priori (se la struttura causale è nota) o appreso congiuntamente con le rappresentazioni. La chiave è che i fattori latenti non sono semplicemente variabili statisticamente indipendenti (come nel VAE standard) ma causalmente strutturati: modificare z_i attraverso un intervento causale produce cambiamenti prevedibili su z_j se j è discendente di i nel grafo. Per Var-JEPA, l'estensione CausalVAE richiederebbe di strutturare il DAG tra (s_x, z, s_y) in modo esplicito come un grafo causale, specificando che s_y è causalmente discendente di s_x e z (ma non viceversa).

DEAR (Disentanglement with Explicit Association of Representations) affronta il problema del disentanglement causale attraverso un termine di regolarizzazione che penalizza le correlazioni tra fattori latenti che non sono previste dalla struttura causale nota. L'idea è analoga alla KL-regularizzazione di Var-JEPA ma applicata non rispetto a un prior gaussiano standard bensì rispetto alla distribuzione causale target $P(z_1, ..., z_d) = \Pi_i P(z_i \mid Pa(z_i))$. Per Var-JEPA multimodale (con viste di testo, immagini e dati sensoriali), un'estensione DEAR potrebbe specificare un grafo causale inter-modale --- ad esempio, la descrizione testuale causa la rappresentazione visiva e non viceversa --- e regolarizzare le rappresentazioni per rispettare questa struttura. Questo permetterebbe di incorporare conoscenza causale a priori sul dominio applicativo direttamente nell'obiettivo di apprendimento, senza richiedere supervisione causale esplicita.

Capitolo 16 — Memoria Episodica e Semantica nei Modelli Linguistici: Architetture Ibride

16.1 La Memoria come Condizione della Conoscenza Cumulativa

Tra i quattro pilastri fondazionali che Mumuni e Mumuni (2025) identificano come condizioni necessarie per l'intelligenza artificiale generale, la memoria occupa una posizione funzionalmente distinta dagli altri tre. Mentre embodiment, radicamento simbolico e causalità riguardano la *natura* della conoscenza — come essa si acquisisce, si ancora e si struttura in relazioni causali — la memoria riguarda la sua *persistenza e accessibilità nel tempo*. In termini cognitivi, la memoria non genera conoscenza fondamentalmente nuova sul mondo, ma riorganizza quella già acquisita per renderla disponibile ai compiti cognitivi futuri.

Il meccanismo di memoria differisce fondamentalmente dagli altri processi cognitivi esaminati nelle sezioni precedenti: esso svolge principalmente un ruolo di preservazione, consolidamento e successiva disponibilità della conoscenza importante acquisita attraverso questi altri processi. Il ruolo principale della memoria nell'AGI è ricostruire e organizzare la conoscenza già ottenuta per compiti cognitivi ad alto livello e per immagazzinarla per un riutilizzo futuro.

La memoria facilita l'apprendimento continuo (*continual learning*) o l'apprendimento a vita (*lifelong learning*), un tratto importante dell'intelligenza biologica. Essa può anche servire come mezzo per incorporare conoscenza a priori nel sistema di IA. Un meccanismo di memoria efficiente consente anche di aggirare calcoli costosi riutilizzando variabili cognitive e soluzioni già calcolate: quando gli esseri umani apprendono per la prima volta un nuovo compito come guidare, richiede attenzione costante e sforzo consapevole; ma dopo aver interiorizzato la competenza richiesta in memoria attraverso la pratica costante, i compiti appresi possono essere svolti senza sforzo senza molta attenzione.

16.2 Tassonomia dei Sistemi di Memoria

La psicologia cognitiva e le neuroscienze identificano tre grandi classi di sistemi di memoria, ciascuna con caratteristiche funzionali, capacità e durata di ritenzione distinte. Questa tassonomia, ben consolidata nella letteratura sulla cognizione biologica, costituisce il punto di riferimento principale per le implementazioni artificiali.

Memoria sensoriale. Il meccanismo di memoria sensoriale è la fase iniziale nel processo di elaborazione delle informazioni ed è associato principalmente alla registrazione delle sensazioni. Il suo ruolo è registrare gli input percettivi dai vari sensori e sistemi di input dell'agente, e fungere da filtro per la vasta quantità di informazioni sensoriali continue dall'ambiente, focalizzandosi sulle informazioni più salienti mentre vengono ignorati i segnali rumorosi. In termini di durata di archiviazione, questo tipo di memoria conserva le informazioni per il tempo più breve, rendendole disponibili ai sistemi sensoriali solo abbastanza a lungo da superare l'inerzia intrinseca associata ai sistemi di elaborazione sensoriale.

Formalmente, la memoria sensoriale può essere modellata come un buffer a capacità fissa con politica di scadenza temporale:

$$B(t) = \{(o_i, t_i) : t - t_i \leq \Delta\tau\}$$

dove o_i è l'osservazione acquisita al tempo t_i, t è il tempo corrente, e $\Delta\tau$ è la finestra di ritenzione temporale. Nei sistemi ciber-fisici che utilizzano sensori per la percezione, questo meccanismo è implementato nativamente attraverso buffer I/O con latching e meccanismi di buffering dei dati.

Memoria di lavoro. La memoria di lavoro — o memoria a breve termine — mantiene quantità relativamente piccole di informazioni in uno stato attivo e facilmente accessibile per durate relativamente brevi durante o per l'elaborazione per compiti cognitivi immediati, inclusa percezione, processo decisionale, ragionamento, esecuzione di istruzioni e funzioni esecutive. Il concetto, introdotto da Baddeley, concettualizza la memoria a breve termine come un blocco di memoria in cui l'informazione necessaria per l'esecuzione dei compiti correnti circola.

Nel contesto dei LLM, la finestra contestuale svolge una funzione analoga alla memoria di lavoro. Il modello mantiene i token più recenti nella finestra contestuale, ricordando solo ciò che è stato presentato poche frasi prima del presente. Il contenuto, simile alla memoria di lavoro, decade facilmente o viene sovrascritto da nuovi input. La dimensione della finestra contestuale determina la capacità di questa memoria. Molti lavori utilizzano precisamente la finestra contestuale del LLM come meccanismo di memoria a breve termine, sfruttando le informazioni nel contesto come descrizione dello stato, del compito o dell'obiettivo intermedio.

Memoria a lungo termine. La memoria a lungo termine preserva le informazioni per periodi di tempo estesi. Nei sistemi cognitivi biologici, la memoria a lungo termine può conservare informazioni per l'intera vita dell'organismo, sebbene nel tempo le informazioni possano essere soggette a decadimento, distorsione o perdita. Nei sistemi sintetici, i problemi di perdita di informazioni possono essere evitati incorporando schemi espliciti che mantengono le informazioni permanentemente.

16.3 La Memoria Semantica: Architetture e Implementazioni

La memoria semantica mantiene conoscenza generale che non dipende da contesti specifici o esperienze "personali" o uniche dell'agente. La conoscenza codificata nella memoria semantica consiste in fatti, formule, regole e leggi generali, definizioni, nonché parole e simboli con i loro significati. La memoria semantica si concentra sulla conoscenza concettuale e di alto livello del mondo e su come questa sia espressa in termini di simboli, immagini e parlato.

Nei LLM, la conoscenza semantica è catturata naturalmente durante l'addestramento. Nel corso dell'addestramento, i modelli linguistici multimodali imparano ad associare parole, frasi, immagini, simboli e concetti sulla base dei pattern statistici nei dati di addestramento. Questo processo consente al LLM di costruire una ricca rappresentazione interna della conoscenza generale, mantenuta nella memoria a lungo termine e richiamata in futuro per supportare compiti cognitivi.

Questa forma di memoria semantica distribuita nei parametri del modello può essere formalizzata come segue. Sia $\theta \in \mathbb{R}^M$ il vettore dei parametri del modello, addestrato su un corpus $\mathcal{D} = \{(x_i, y_i)\}$ mediante minimizzazione della funzione di perdita:

$$\theta^* = \text{argmin}_\theta \mathcal{L}(\theta) = \text{argmin}_\theta -(1/|\mathcal{D}|) \sum_{(x,y)\in\mathcal{D}} \log P_\theta(y \mid x)$$

Il vettore θ^* codifica implicitamente la conoscenza semantica estratta dal corpus. L'interrogazione di questa conoscenza — ovvero il richiamo della memoria semantica — avviene attraverso la computazione forward pass:

$$y = P_{\theta^*}(\cdot \mid x_{query})$$

Questo meccanismo di memoria ha il vantaggio della compattezza e dell'efficienza computazionale ma presenta due limiti strutturali. Il primo è il *catastrophic forgetting*: l'aggiornamento dei parametri per incorporare nuova conoscenza altera la conoscenza preesistente in modo imprevedibile. Il secondo è la staticità: una volta completato l'addestramento, la memoria semantica parametrica non può essere aggiornata senza un nuovo ciclo di addestramento.

Tecniche come *elastic weight consolidation* (EWC) tentano di mitigare il catastrophic forgetting penalizzando le modifiche ai parametri ritenuti critici per la conoscenza preesistente:

$$\mathcal{L}_{EWC}(\theta) = \mathcal{L}_{new}(\theta) + (\lambda/2) \sum_i F_i (\theta_i - \theta^*_i)^2$$

dove $F = \{F_1, \ldots, F\}$ è la diagonale della matrice di informazione di Fisher calcolata sul compito precedente — una stima dell'importanza di ciascun parametro per la conoscenza precedentemente acquisita.

La memoria semantica può anche essere implementata in forma strutturata, usando grafi della conoscenza e modelli grafici causali come memoria a lungo termine nei LLM. Queste strutture di conoscenza specializzate all'interno dei LLM possono immagazzinare fatti, regole, concetti e relazioni in modo persistente, consentendo ai LLM di recuperare informazioni utili al bisogno.

16.4 La Memoria Episodica: Struttura e Implementazione

La memoria episodica mantiene informazioni su eventi importanti, esperienze e informazioni contestuali associate — ovvero informazioni sui tempi, luoghi, contesto ambientale circostante e natura degli eventi. Questa informazione è rappresentata come sequenza ordinata nel tempo di esperienze. Le informazioni non sono additive: diversi eventi registrati sono separati e le esperienze non sono generalizzate né accumulate.

A differenza della memoria semantica, che cattura regolarità statistiche attraverso grandi insiemi di dati, la memoria episodica è indicizzata da coordinate spazio-temporali specifiche: ogni traccia mnestica episodica è associata a *quando* e *dove* l'evento è accaduto, insieme alla sua natura qualitativa. Questa struttura consente l'introspezione e la riflessione sul passato nel contesto della realtà attuale, fornendo così un mezzo di auto-miglioramento.

Formalmente, un episodio può essere rappresentato come una tupla:

$e = \langle \tau, l, s, m, c \rangle$

dove τ è la marcatura temporale, l è il contesto spaziale (luogo), s è lo stato del sistema all'inizio dell'episodio, m è la sequenza di osservazioni-azioni durante l'episodio, e c è il contesto contestuale complessivo (agenti presenti, stati emotivi rilevanti, informazioni sensoriali associate). L'insieme degli episodi memorizzati costituisce una memoria episodica $E = \{e_1, e_2, \ldots, eN\}$.

Il recupero episodico in risposta a una query q si configura come un problema di recupero semantico nella memoria strutturata:

$e^* = \text{argmax}_\{e \in E\} \text{sim}(\text{Enc}(q), \text{Enc}(e))$

dove Enc è un encoder che proietta sia le query che gli episodi in uno spazio metrico condiviso, e sim è una funzione di similarità. Nelle architetture pratiche, i *vector databases* consentono recupero efficiente tramite ricerca approssimata del vicino più prossimo (ANN), con complessità sublineare nel numero degli episodi memorizzati attraverso strutture dati come HNSW (Hierarchical Navigable Small World graphs).

16.5 Il Meccanismo di Attenzione come Memoria di Lavoro

Mentre gran parte della letteratura recente si concentra sulla memoria esterna esplicita, è importante riconoscere che i LLM basati su Transformer implementano una forma sofisticata di memoria di lavoro attraverso il meccanismo di attenzione. La comprensione di questo meccanismo è cruciale per qualsiasi tentativo di potenziare le capacità mnesiche di questi sistemi.

Il meccanismo di self-attenzione in un singolo strato Transformer opera come segue. Dato un input $X \in \mathbb{R}^{L \times d}$ di L token con dimensione d, si calcolano tre matrici:

$Q = X W_Q, K = X W_K, V = X W_V$

dove $W_Q, W_K, W_V \in \mathbb{R}^{d \times d_k}$ sono matrici di proiezione apprese. L'output dell'attenzione è:

$\text{Attn}(Q, K, V) = \text{softmax}(Q K^T / \sqrt{d_k}) V$

La matrice di attenzione $A = \text{softmax}(Q K^T / \sqrt{d_k}) \in \mathbb{R}^{L \times L}$ codifica la struttura associativa della memoria di lavoro: l'entrata A_{i}□ quantifica quanto il token i debba attendere al token j per produrre la propria rappresentazione contestualizzata. Questa struttura è, in modo profondo, una forma di memoria associativa capacitiva: l'output per ciascun token è una combinazione lineare delle *values* di tutti i token precedenti (in self-attenzione causale), pesata dalla rilevanza computata dinamicamente.

La complessità quadratica $O(L^2)$ del meccanismo di attenzione limita la finestra contestuale utilizzabile. Per sequenze molto lunghe, questo introduce il cosiddetto *lost in the middle problem*:

ricerche recenti mostrano che i modelli mostrano spesso bias per le informazioni all'inizio e alla fine della finestra contestuale, mettendo in secondo piano le informazioni nel mezzo. Questa limitazione — dove una grande porzione di informazioni al di fuori delle due estremità non viene memorizzata — ha motivato lo sviluppo di meccanismi di memoria a lungo termine esplicita.

Per l'attenzione multi-testa (*multi-head attention*), H teste parallele operano in sotto-spazi proiettati distinti:

MultiHead(Q, K, V) = Concat(head_1, …, head) W_O

head_i = Attn(Q W_Q^i, K W_K^i, V W_V^i)

dove W_O ∈ ℝ^{hd_v × d} è la proiezione di output. Questo consente al modello di attendere a informazioni provenienti da rappresentazioni distinte nello spazio in posizioni diverse: le diverse teste catturano aspetti diversi delle relazioni tra i token — alcune teste catturano relazioni sintattiche, altre relazioni semantiche, altre ancora dipendenze a lungo raggio.

16.6 Memoria Esplicita e Vector Databases

Per affrontare le limitazioni della memoria implicita nei parametri e nella finestra contestuale, approcci più recenti hanno proposto di consentire archiviazione selettiva di informazioni persistenti e contestualmente rilevanti in memoria informatica per il riutilizzo. In particolare, conoscenza specifica del dominio può essere archiviata esplicitamente in memoria come conoscenza a priori per integrare l'estesa conoscenza generica appresa dal LLM.

Tra le implementazioni pratiche, i *vector databases* si sono imposti come soluzione preferita. Un vector database D è una struttura dati che archivia coppie (k, v) dove $k \in \mathbb{R}^d$ è un vettore-chiave ottenuto codificando il documento d mediante un encoder Enc_K, e v è il contenuto del documento stesso. Il recupero per similarità dato una query q avviene come:

$\mathcal{D}$_q = {(k_i, v_i) : cos(Enc_Q(q), k_i) ≥ τ} = {(k_i, v_i) : (Enc_Q(q) · k_i)/(‖Enc_Q(q)‖₂ ‖k_i‖₂) ≥ τ}

dove τ è una soglia di similarità e Enc_Q è un encoder (potenzialmente diverso da Enc_K) per le query. In pratica, l'esatta ricerca del vicino più prossimo è sostituita da algoritmi ANN come HNSW o IVF-PQ, con complessità media O(log N) per N documenti archiviati.

Questo sistema di memoria ha il vantaggio di consentire non solo un recupero rapido, ma anche manipolazioni sofisticate e granulari a livello di feature. Inoltre, questo metodo di rappresentazione rende semplice l'uso di operazioni apprese in fase di addestramento invece che basarsi su routine analitiche predefinite per manipolare le informazioni archiviate.

L'integrazione di vector databases con LLM produce architetture ibride del tipo MemGPT, in cui il modello gestisce attivamente l'allocazione tra memoria di lavoro (finestra contestuale) e memoria a lungo termine esterna, orchestrando operazioni di lettura e scrittura sulla memoria persistente esattamente come un sistema operativo gestisce la gerarchia di memoria tra cache, RAM e disco.

16.7 La Memoria Procedurale e il Suo Ruolo nell'Autonomia

La memoria procedurale riguarda l'acquisizione, la conservazione e il richiamo della conoscenza relativa ai passi logici necessari per svolgere attività complesse. Questo implica comunemente abilità motorie come guida autonoma, cucina e manipolazione robotica. Nel campo dei modelli linguistici di grande scala, la memoria procedurale facilita funzioni cognitive come la pianificazione delle attività, il seguire istruzioni, il ragionamento e l'esecuzione di azioni fisiche.

Nei sistemi cognitivi biologici, un aspetto principale della memoria procedurale riguarda processi subconsci — processi che si verificano al di fuori della consapevolezza dell'agente, come il priming e il condizionamento classico. Il compito consiste nell'apprendere e immagazzinare associazioni naturali tra stimoli e risposte corrispondenti, consentendo in tal modo di invocare automaticamente risposte appropriate nelle situazioni giuste, aggirando la necessità di eseguire calcoli cognitivi complessi richiesti per il ragionamento.

Nei LLM, la conoscenza procedurale è comunemente appresa implicitamente e immagazzinata nella memoria a lungo termine, per virtù della capacità dei LLM di apprendere sequenze di azioni, strutture e relazioni nei dati di addestramento. Dopo la fase di pre-addestramento, i LLM possono essere ulteriormente messi a punto su dataset sensorimotori specifici del compito, che di solito includono istruzioni dettagliate passo-passo su come svolgere i compiti target. Il processo di fine-tuning mira all'interiorizzazione o al raffinamento della capacità del modello di generare e seguire piani di esecuzione espliciti necessari per risolvere il problema dato.

Formalmente, la memoria procedurale per un compito τ può essere rappresentata come una funzione di condizionamento:

$\pi_\tau(a_t \mid s_t, \tau, h_\{<t\})$

dove s_t è lo stato corrente, τ è la specifica del compito, $h_\{<t\} = \{(s_1,a_1), \ldots, (s_\{t-1\}, a_\{t-1\})\}$ è la storia delle azioni-osservazioni precedenti, e a_t è l'azione da eseguire. L'apprendimento di questa policy costituisce l'interiorizzazione della memoria procedurale: il sistema non calcola più consapevolmente ogni passo ma richiama direttamente la sequenza appropriata dato il contesto.

16.8 Retrieval-Augmented Generation come Memoria Esterna

Il Retrieval-Augmented Generation (RAG) rappresenta una forma di memoria esterna che emula la capacità umana di consultare risorse di conoscenza esterne. Uno dei vantaggi chiave che gli esseri umani hanno sugli altri animali è la capacità di usare risorse di conoscenza esterna e strumenti per aumentare o estendere le proprie competenze — per esempio leggendo manuali, libri, o navigando in internet per le informazioni necessarie per un dato compito.

L'architettura RAG si configura come un sistema a due componenti: un retriever R e un generatore G. Il retriever seleziona i documenti rilevanti da una base di conoscenza $\mathcal{K}$:

$$\mathcal{D}_q = R(q; \mathcal{K}) = \text{top-k}\{d \in \mathcal{K} : \text{score}(q, d)\}$$

Il generatore produce l'output condizionato sulla query e sui documenti recuperati:

$$\hat{y} = G(q, \mathcal{D}_q) = P_\theta(\hat{y} \mid q, d_1, \ldots, d_k)$$

In questa architettura, la funzione di scoring nel retriever è tipicamente:

$$\text{score}(q, d) = \text{Enc}_Q(q)^\top \text{Enc}_D(d) \,/\, (\|\text{Enc}_Q(q)\|_2 \, \|\text{Enc}_D(d)\|_2)$$

dove Enc_Q e Enc_D possono essere encoder densi appresi congiuntamente (*dense retrieval*) o separati (*bi-encoders*). Varianti più sofisticate come RAG-Fusion generano molteplici riformulazioni della query e poi aggregano i risultati del recupero tramite Reciprocal Rank Fusion:

$$\text{RRF_score}(d) = \sum_{q' \in Q_\text{reformulations}} 1/(k + \text{rank}(d, q'))$$

dove k è un parametro di ammortizzazione e rank(d, q') è il rango del documento d nei risultati per la riformulazione q'.

Il principale svantaggio di questo approccio è che le informazioni esterne possono non essere garantite come corrette a causa di errori sconosciuti, inconsistenze o informazioni incomplete. Può anche verificarsi una perdita completa di accesso alle informazioni esterne per vari motivi; e le informazioni esterne possono essere maggiormente esposte a utenti malintenzionati e attori malevoli, mettendo in pericolo la sicurezza.

16.9 Verso Architetture di Memoria Integrate

Il panorama degli approcci alla memoria artificiale nei LLM rivela una struttura a complementarità reciproca. I diversi sistemi di memoria che la cognizione biologica integra — sensoriale, di lavoro, semantica, episodica, procedurale — non sono componenti isolate ma parti di un'architettura unificata in cui le informazioni fluiscono bidirezionalmente tra di loro.

Nel framework di Mumuni e Mumuni, l'architettura di memoria ideale per un sistema AGI deve soddisfare almeno quattro requisiti. Primo, deve essere *scalabile*: la capacità di archiviazione non può essere limitata dalle dimensioni del modello o dalla lunghezza della finestra contestuale. Secondo, deve essere *dinamica*: il sistema deve poter acquisire, consolidare e dimenticare selettivamente le informazioni in risposta all'esperienza continua. Terzo, deve essere *strutturata*: non ogni informazione ha lo stesso grado di generalità; alcune tracce sono episodiche e legate a contesti specifici, altre sono semantiche e generalizzano tra contesti. Quarto, deve essere *accessibile in modo contestualmente sensibile*: il recupero deve essere guidato non da semplici criteri di similarità superficiale ma da valutazioni pragmatiche di rilevanza rispetto agli obiettivi e ai sotto-obiettivi correnti dell'agente.

Le architetture ibride più recenti — come MemGPT, HippoRAG e AriGraph — tentano di approssimare questi requisiti combinando memoria parametrica, finestra contestuale come working memory, e vector databases come memoria a lungo termine episodica e semantica. L'elemento ancora mancante è un meccanismo di consolidamento robusto, analogo al processo ippocampale-neocorticale della cognizione biologica, capace di trasferire in modo selettivo le tracce dalla memoria a breve termine a quella a lungo termine, generalizzando le regolarità e preservando le particolarità significative. Questo rimane un problema aperto che richiederà integrazioni architetturali che vadano al di là delle soluzioni ingegneristiche attualmente disponibili.

16.10 Memorie differenziabili e architetture di memoria per sistemi JEPA

L'integrazione di meccanismi di memoria differenziabili con le architetture JEPA è una direzione di ricerca che promette di combinare la qualità delle rappresentazioni latenti prodotte da JEPA con la capacità di memorizzazione e recupero selettivo necessaria per l'intelligenza generale. I progressi recenti nei Differentiable Neural Computers (DNC, Graves et al. 2016), nelle Neural Turing Machines (NTM, Graves et al. 2014) e nelle Memory-Augmented Neural Networks (MANN) forniscono i componenti architetturali per questa integrazione.

Il DNC implementa una memoria esterna M ∈ ℝ^{N×W} (N slot di memoria, ciascuno di dimensione W) con meccanismi di lettura e scrittura differenziabili controllati da un controller neurale. La scrittura in memoria è: M_t[i] = M_{t-1}[i] · (1 − w_t^w[i] · e_t^⊤) + w_t^w[i] · a_t^⊤, dove w_t^w è il vettore di pesi di scrittura, e_t è il vettore di cancellazione e a_t è il vettore da scrivere. La lettura è: r_t = Σ_i w_t^r[i] · M_t[i], dove w_t^r è il vettore di pesi di lettura. L'integrazione di questo meccanismo con Var-JEPA potrebbe essere realizzata come segue: le rappresentazioni latenti s_x prodotte dal context encoder vengono scritte in memoria durante la sequenza di addestramento, e vengono recuperate durante l'inferenza per arricchire il prior condizionale p_θ(s_y | s_x, z, M). In questo schema, M funge da memoria episodica che preserva le rappresentazioni di esempi passati, migliorando la predizione per campioni simili a quelli già visti.

Un'architettura più recente e scalabile è quella dei Retrieval-Augmented Transformers (RAT), che combinano l'attenzione del transformer con il recupero da una memoria esterna indicizzata per similarità. Nel contesto di LLM-JEPA, un RAT potrebbe recuperare automaticamente le K coppie (Text, Code) più simili alla coppia corrente dalla memoria e usare le loro rappresentazioni come contesto aggiuntivo per il predittore. Il recupero avviene attraverso il termine cosine similarity nel latent space: Top-K({Enc(Text_i)}, Enc(Text_query)), e le rappresentazioni recuperate vengono incorporate come token aggiuntivi nel predittore. Questo produce un sistema LLM-JEPA con memoria episodica esplicita, capace di migliorare le predizioni per query simili a esempi già incontrati senza riaddestrare il modello.

16.11 Apprendimento continuo nella prospettiva variazionale: evitare il catastrophic forgetting con Var-JEPA

Il catastrophic forgetting --- la tendenza delle reti neurali ad "dimenticare" conoscenze acquisite su compiti precedenti quando vengono addestrate su nuovi compiti --- è uno dei problemi più fondamentali nell'apprendimento continuo (continual learning). La struttura variazionale di Var-JEPA offre una prospettiva naturale su questo problema e suggerisce approcci di regolarizzazione principiati che vanno al di là delle tecniche euristiche attualmente prevalenti.

Il framework Elastic Weight Consolidation (EWC, Kirkpatrick et al. 2017) protegge i parametri critici per compiti precedenti penalizzando le modifiche ai pesi che hanno alta Fisher information per quei compiti: $\mathcal{L}_EWC = \mathcal{L}_new + (\lambda/2)\ \Sigma_i\ F_i(\theta_i - \theta_i^*)^2$, dove F_i è la i-esima voce della diagonale della Fisher information matrix e θ_i* sono i parametri ottimali per il compito precedente. Per Var-JEPA, la Fisher information matrix può essere calcolata rispetto all'ELBO sul dataset del compito precedente, fornendo un segnale di importanza parametrica più principiato rispetto all'EWC standard applicato alla cross-entropy. I parametri con alta importanza ELBO sono quelli che determinano la struttura delle distribuzioni variazionali --- in particolare i pesi del prior condizionale appreso p_θ(s_y | s_x, z) che codificano le relazioni strutturali tra contesto e target --- e che dovrebbero essere protetti dall'aggiornamento su nuovi compiti.

Un approccio complementare al continual learning variazionale è il framework CURL (Continual Unsupervised Representation Learning, Rao et al. 2019), che usa un modello generativo bayesiano con componenti miste per modellare la distribuzione dei dati di task diversi. In CURL, i nuovi task vengono gestiti aggiungendo nuove componenti alla miscela senza modificare quelle esistenti, mentre la selezione del componente appropriato avviene attraverso inferenza bayesiana. Per Var-JEPA, un analogo di CURL richiederebbe di mantenere una "biblioteca" di prior condizionali {p_θ^{(t)}(s_y | s_x, z)}_{t=1}^T --- uno per ogni compito precedente --- e selezionare il prior appropriato attraverso un meccanismo di attention condizionale al task context. Questo schema è computazionalmente costoso ma garantisce che la conoscenza dei compiti precedenti non venga dimenticata, poiché le loro distribuzioni variazionali sono preservate esplicitamente nella biblioteca.

Capitolo 17 — Confronto con il Framework AMI di LeCun: Convergenze e Divergenze

17.1 Premessa epistemologica: due tradizioni a confronto

Qualsiasi tentativo di valutare con rigore le prospettive sull'intelligenza artificiale generale che emergono dalla letteratura recente deve misurarsi con un dato strutturale difficile da eludere: le tradizioni intellettuali che animano il dibattito attuale non condividono né le stesse premesse ontologiche sull'intelligenza né le medesime intuizioni circa il percorso più promettente verso sistemi davvero generalisti. Il framework dell'Autonomous Machine Intelligence elaborato da Yann LeCun — esposto nella sua forma più sistematica nel documento programmatico del 2022 intitolato *A Path Towards Autonomous Machine Intelligence* — e la prospettiva LLM-centrica articolata da Alhassan Mumuni e Fuseini Mumuni nel survey del 2025 sui principi fondazionali dell'AGI con i Large Language Models rappresentano, in questo senso, due orientamenti che si parlano raramente in modo diretto, che divergono su questioni architetturali profonde, ma che condividono, a un livello di astrazione sufficiente, una preoccupazione comune: quella di dotare le macchine di una forma di intelligenza genuinamente flessibile, robusta e capace di operare nel mondo reale senza richiedere ogni volta una supervisione umana minuziosa.

La tensione fra questi due paradigmi non è meramente tecnica. Essa riflette una divergenza circa la natura stessa della cognizione e il modo in cui essa si relaziona al linguaggio, alla percezione sensorimotoria e alla struttura latente del mondo. LeCun muove da una critica radicale ai modelli autoregressivi generativi, sostenendo che la predizione nello spazio dell'input — ovvero la generazione token per token di testo o pixel per pixel di immagini — è fondamentalmente un obiettivo mal posto, perché costringe il modello a codificare un'enorme quantità di dettagli irrilevanti anziché apprendere le strutture astratte e causali che governano il comportamento del mondo. Mumuni e Mumuni, per converso, riconoscono nei Large Language Models un substrato promettente per l'AGI, a patto di arricchirlo con quattro principi cognitivi che oggi mancano o sono soltanto abbozzati: il radicamento corporeo (*embodiment*), il fondamento simbolico (*symbol grounding*), la causalità e la memoria. La dialettica fra queste posizioni è feconda proprio perché nessuna delle due è riducibile a un semplicismo ingenuo, e perché i risultati empirici dei due paper tecnici qui considerati — LLM-JEPA di Huang, LeCun e Balestriero (2025) e Var-JEPA di Gögl e Yau (2026) — suggeriscono che le distanze, pur reali, sono forse meno invalicabili di quanto la retorica programmatica di ciascuna tradizione lasci supporre.

17.2 L'architettura AMI: struttura modulare e principi fondanti

La proposta di LeCun per un'intelligenza artificiale autonoma non si configura come un singolo algoritmo o un'unica architettura neurale, bensì come un sistema modulare composto da sotto-sistemi specializzati che interagiscono in modo coordinato. La comprensione di questo sistema richiede di tenere insieme tre livelli di analisi: quello funzionale, che descrive *cosa* ciascun modulo deve fare; quello architetturale, che specifica *come* i moduli sono connessi e in quale ordine si attivano; e quello matematico, che formalizza gli obiettivi di apprendimento e i flussi informativi.

Al centro del framework AMI si trova il **modello del mondo** (*world model*), che LeCun identifica come il componente essenziale di qualsiasi forma di intelligenza genuinamente autonoma. Il modello del mondo ha il compito di predire, in forma astratta, le conseguenze delle azioni dell'agente e gli stati futuri dell'ambiente, senza dover ricostruire ogni dettaglio

sensoriale della realtà. Questa scelta — predire in uno spazio latente astratto piuttosto che nello spazio dell'osservazione grezza — è la chiave di volta che distingue il framework AMI dagli approcci generativi classici basati su autoencoders variazionali (VAE) o modelli di diffusione.

Attorno al modello del mondo, il framework AMI dispone cinque ulteriori componenti. Il **configuratore** riceve gli obiettivi di alto livello provenienti dall'esterno o dall'agente stesso e ha il compito di parametrizzare tutti gli altri moduli in funzione del compito corrente: in termini matematici, il configuratore può essere pensato come una funzione γ che mappa una specifica di obiettivo $\mathcal{G}$ in un insieme di parametri contestuali $\{\gamma_1, \gamma_2, ..., \gamma\}$ che modulano il comportamento di ciascun sotto-sistema. Il **modulo percettivo** trasforma gli input sensoriali grezzi — immagini, audio, testo, segnali propriocettivi — in rappresentazioni latenti che fungono da input per il modello del mondo. Il **modulo dei costi** assegna scalari di costo alle situazioni e ai piani, consentendo all'agente di valutare la desiderabilità degli stati futuri in modo differenziabile: questo modulo è profondamente distinto dalla funzione di ricompensa dell'apprendimento per rinforzo classico, perché non è binario o sparso ma strutturato e composto da termini che catturano sia il progresso verso l'obiettivo sia il rispetto di vincoli di sicurezza. La **memoria a breve termine** mantiene in forma attiva le rappresentazioni pertinenti al compito corrente per tutta la durata di un episodio. Infine, l'**attore** traduce i piani computati dal modello del mondo in sequenze di azioni concrete.

La novità più radicale di questo framework, sul piano dell'apprendimento automatico, risiede nell'obiettivo con cui il modello del mondo viene addestrato. LeCun propone di abbandonare la predizione nello spazio dell'input — cioè la ricostruzione pixel-per-pixel o token-per-token — in favore di un obiettivo che opera interamente nello spazio delle rappresentazioni latenti. Questo obiettivo è precisamente ciò che viene chiamato **Joint Embedding Predictive Architecture (JEPA)**.

17.3 JEPA come nucleo matematico del framework AMI

La formulazione originale di JEPA prende le mosse da una coppia di osservazioni (x, y) legate da una relazione strutturale — per esempio, due ritagli spaziali diversi della stessa immagine, due istanti temporali successivi di un video, o due viste generate da aumentazioni diverse. Due encoders — il codificatore contestuale f^{ctx} e il codificatore target f^{trg} — proiettano rispettivamente x e y in rappresentazioni latenti:

$$s_x = f^{ctx}(x), \quad s_\gamma = f^{trg}(y)$$

Un predittore g prende in input la rappresentazione contestuale s_x e una variabile ausiliaria z — che ha il compito di catturare la variabilità irriducibile che non può essere inferita dal solo contesto — e produce una stima della rappresentazione target:

$$\hat{s}_\gamma = g(s_x, z)$$

L'obiettivo di apprendimento minimizza la distanza fra la stima e la rappresentazione target vera:

$$\mathcal{L}_\text{JEPA} = \mathbb{E}[\|\hat{s}_\gamma - \text{sg}(s_\gamma)\|^2_2]$$

dove sg(·) denota l'operazione di *stop-gradient*, che impedisce la propagazione dei gradienti attraverso il codificatore target. Questa asimmetria è cruciale: se si permettesse ai gradienti di fluire liberamente attraverso entrambi gli encoders, il sistema collasserebbe verso la soluzione banale in cui tutte le rappresentazioni convergono a un vettore costante, annullando la distanza ma rendendo le rappresentazioni inutili per qualsiasi compito a valle. L'operazione di stop-gradient, unitamente all'aggiornamento del codificatore target tramite media mobile esponenziale (EMA) dei parametri del codificatore contestuale:

$$\theta^{\text{trg}} \leftarrow \tau \cdot \theta^{\text{trg}} + (1 - \tau) \cdot \theta^{\text{ctx}}, \quad \tau \in [0, 1)$$

garantisce che le rappresentazioni mantengano diversità informativa senza richiedere termini di regolarizzazione espliciti basati sulla varianza o sulla covarianza.

Ciò che rende JEPA particolarmente interessante dal punto di vista teorico — e che il paper di Gögl e Yau (2026) analizza con notevole profondità — è la sua relazione con i modelli generativi a variabile latente. Gögl e Yau dimostrano che il design canonico di JEPA — due encoders accoppiati con un predittore dal contesto al target — non è altro che una specializzazione deterministica di un modello a variabile latente accoppiata (coupled VAE), nella quale la regolarizzazione è imposta tramite euristiche architetturali (stop-gradient, EMA) piuttosto che attraverso un'esplicita funzione di verosimiglianza. Il loro argomento principale — che la separazione retorica fra JEPA e modellazione generativa probabilistica è largamente illusoria sul piano strutturale — apre una prospettiva nuova sulla natura dell'obiettivo AMI, e su cui si dovrà tornare con attenzione nella sezione dedicata alle convergenze.

Il processo generativo latente che Gögl e Yau reverse-engineerano a partire dall'architettura JEPA fattorizza la distribuzione congiunta sulle osservazioni e sulle variabili latenti come segue:

$$p_\theta(x, y, s_x, z, s_\gamma) = p(s_x) \cdot p(z) \cdot p_\theta(x \mid s_x) \cdot p_\theta(s_\gamma \mid s_x, z) \cdot p_\theta(y \mid s_\gamma)$$

dove $p(s_x) = \mathcal{N}(s_x; 0, I)$ e $p(z) = \mathcal{N}(z; 0, I)$ sono prior standard gaussiani sulle variabili latenti, mentre $p_\theta(s_\gamma \mid s_x, z)$ è il **prior condizionale appreso**, che funge precisamente da rete predittore del JEPA originale nella sua veste generativa. L'Evidence Lower Bound associato a questo modello generativo, derivato mediante la diseguaglianza di Jensen applicata alla log-verosimiglianza marginale $\log p_\theta(x, y)$, si articola in cinque termini interpretativamente distinti:

$$\begin{aligned}\text{ELBO}(x, y; \theta, \phi) = {} & \mathbb{E}_{q_\phi(s_x \mid x)}[\log p_\theta(x \mid s_x)] \\ & + \mathbb{E}_{q_\phi(s_x, z, s_\gamma \mid x, y)}[\log p_\theta(y \mid s_\gamma)] \\ & - \text{KL}(q_\phi(s_x \mid x) \,\|\, p(s_x))\end{aligned}$$

$- \mathrm{KL}(q_\phi(z \mid s_x) \,\|\, p(z))$
$- \mathrm{KL}(q_\phi(s_\gamma \mid s_x, z, y) \,\|\, p_\theta(s_\gamma \mid s_x, z))$

Il primo termine misura la qualità della ricostruzione del contesto x a partire dalla sua rappresentazione latente s_x. Il secondo termine misura la qualità della generazione del target y a partire dalla rappresentazione latente del target s_γ. I tre termini di divergenza di Kullback-Leibler regolarizzano rispettivamente la distribuzione posteriore del contesto verso il prior standard, la distribuzione posteriore della variabile ausiliaria verso il suo prior standard, e la distribuzione posteriore del target verso il prior condizionale appreso — che è esattamente il predittore di JEPA in veste generativa.

Questa formalizzazione variazionale ha un'implicazione teorica diretta che il paper di Gögl e Yau articola con cura: il collasso delle rappresentazioni, che nel JEPA canonico viene prevenuto mediante artifici architetturali come stop-gradient e EMA, nel Var-JEPA è prevenuto *strutturalmente* dai termini di divergenza KL, che esercitano una pressione regolarizzatrice per campione (*per-sample*) verso prior gaussiani standardizzati. In altri termini, la regolarizzazione distribuzionale che il JEPA canonico ottiene per via euristica emerge nell'approccio variazionale come conseguenza necessaria dell'ottimizzazione dell'ELBO.

17.4 Il JEPA applicato ai Large Language Models: un primo punto di contatto

Il paper di Huang, LeCun e Balestriero (2025) rappresenta il tentativo più sistematico finora compiuto di applicare l'obiettivo JEPA ai Large Language Models, colmando quella che gli autori stessi identificano come un'asimmetria ingiustificata fra la ricerca in visione e quella in linguaggio. In visione, gli obiettivi JEPA — a partire dall'I-JEPA (Image JEPA) di Assran et al. (2023) fino al V-JEPA per i dati video di Bardes et al. (2024) — hanno dimostrato benefici empirici significativi rispetto agli obiettivi di ricostruzione nell'input space. Nel dominio del linguaggio naturale, invece, la predizione autoregressiva del token successivo è rimasta pressoché incontrastata come obiettivo di pretraining.

LLM-JEPA propone di combinare l'obiettivo generativo standard degli LLM con un termine JEPA aggiuntivo che opera nello spazio degli embedding. Il presupposto fondamentale è che alcune categorie di dati linguistici offrano nativamente due viste (*views*) della stessa conoscenza sottostante: il testo in linguaggio naturale e il codice che implementa la stessa funzionalità, oppure la descrizione di un'espressione regolare e l'espressione stessa. La perdita composta prende la forma:

$$\mathcal{L}_{\text{LLM-JEPA}} = \sum_{\ell=2}^{L} \mathcal{L}_{\text{LLM}}(\text{Text}_{1:\ell-1}, \text{Text}_{\ell}) + \lambda \cdot d(\text{Pred}(\text{Enc}(\text{Text})), \text{Enc}(\text{Code}))$$

dove il primo termine è la perdita cross-entropica standard per la predizione autoregressiva del token successivo, il secondo è la distanza coseno fra la predizione del codificatore sul testo e la codifica del codice target, $\lambda \geq 0$ è un iperparametro di bilanciamento, e $\mathrm{Pred}(\cdot)$ è il predittore implementato sfruttando l'architettura autoregressiva del modello stesso —

attraverso l'aggiunta di token speciali [PRED] alla fine della sequenza di testo, seguiti dal calcolo dell'embedding dell'ultimo layer.

L'implementazione tecnica di questo obiettivo richiede la gestione della maschera di attenzione in modo che le due viste — testo e codice — non si influenzino reciprocamente nell'encoder, pur essendo processate insieme in un unico forward pass. Gli autori risolvono questo problema con una maschera di attenzione causale a blocchi: all'interno di ciascuna vista, l'attenzione è causale (ogni token può attendere solo i token precedenti della stessa vista); fra le due viste, l'attenzione è completamente mascherata, impedendo ogni flusso informativo diretto.

I risultati empirici di LLM-JEPA, come documentato in dettaglio nel paper, sono statisticamente significativi su quattro dataset (NL-RX-SYNTH, NL-RX-TURK, GSM8K, Spider) e quattro famiglie di modelli (Llama3, Gemma2, OpenELM, OLMo), con miglioramenti dell'accuratezza che variano dal 14% relativo per Llama3 su NL-RX-SYNTH fino al doppio circa per OpenELM sullo stesso dataset. Un ulteriore risultato di rilievo è la resistenza al sovraddattamento (*overfitting*): mentre i modelli addestrati con il solo obiettivo LLM mostrano un degrado delle prestazioni nelle epoche successive alla quarta, i modelli addestrati con $\mathcal{L}$_LLM-JEPA continuano a migliorare fino alla sesta epoca, suggerendo che il termine JEPA esercita un effetto regolarizzatore sulla struttura dello spazio latente.

Questo risultato — il fatto cioè che LeCun stesso sia co-autore di un paper che applica con successo l'obiettivo JEPA ai Large Language Models — è di per sé una testimonianza del fatto che la posizione di LeCun nei confronti degli LLM è più sfumata di quanto la sua retorica critica più nota lasci talvolta supporre. La critica di LeCun non è rivolta ai Transformer come architettura in quanto tale, né ai meccanismi di attenzione, bensì all'obiettivo di pretraining autoregressivo e alla conseguente incapacità di questi modelli di apprendere rappresentazioni genuinamente astratte e causali del mondo.

Capitolo 17 — Parte Seconda: Convergenze e Divergenze

17.5 Le convergenze strutturali: dove i due paradigmi si incontrano

Malgrado la distanza retorica e programmatica che separa il framework AMI dall'approccio LLM-centrico di Mumuni e Mumuni, un'analisi attenta rivela convergenze strutturali non trascurabili, che emergono con particolare nettezza non appena si abbandona il livello del manifesto per scendere al livello dell'architettura concreta.

La prima e più profonda convergenza riguarda il ruolo centrale attribuito al **modello del mondo**. Entrambi i paradigmi riconoscono che un sistema davvero intelligente deve possedere una rappresentazione interna dell'ambiente che gli permetta di anticipare le conseguenze delle proprie azioni e di pianificare sequenze comportamentali su orizzonti temporali estesi. Nel framework AMI, questo imperativo si traduce direttamente

nell'architettura del world model basato su JEPA. Nel framework di Mumuni e Mumuni, la stessa esigenza si manifesta attraverso il concetto di *situational awareness* — la capacità dell'agente di modellare non soltanto lo stato corrente dell'ambiente ma anche la sua evoluzione futura sotto l'effetto delle proprie azioni e di quelle degli altri agenti presenti. La differenza non è nel riconoscimento dell'esigenza, bensì nel modo in cui si propone di soddisfarla: mediante un modello del mondo addestrato con obiettivi predictivi in spazio latente (AMI), o mediante LLM arricchiti di capacità sensorimotorie e meccanismi episodici di memoria (Mumuni e Mumuni).

La seconda convergenza riguarda il **rifiuto della ricostruzione nell'input space come obiettivo sufficiente**. Sebbene Mumuni e Mumuni non discutano esplicitamente la dicotomia JEPA vs. modelli autoregressivi, il loro survey riconosce implicitamente che la conoscenza che i Large Language Models apprendono dai dati è «superficiale e fragile» (*shallow and brittle*) e che le loro capacità cognitive «sono ancora limitate nel senso generalista». Questa diagnosi è strutturalmente congruente con la critica di LeCun: in entrambi i casi si riconosce che la predizione statistica dei pattern nei dati non è equivalente alla comprensione delle strutture causali sottostanti. La differenza, ancora una volta, è nella terapia proposta piuttosto che nella diagnosi del problema.

La terza convergenza, più tecnica, riguarda il trattamento della **memoria**. LeCun, nell'architettura AMI, dedica un modulo separato — la *short-term memory* — alla preservazione delle rappresentazioni rilevanti per l'episodio corrente. Mumuni e Mumuni, nel loro framework per l'AGI con LLM, dedicano un'intera sezione del survey alle diverse tipologie di memoria — sensoriale, di lavoro (*working memory*), semantica, episodica e procedurale — e ai loro meccanismi di implementazione nei sistemi LLM-based. In entrambi i casi, la memoria non è trattata come un semplice buffer di storage ma come un componente cognitivo attivo che interagisce con la percezione, il ragionamento e l'azione in modo bidirezionale: le esperienze passate informano la percezione presente, e la percezione presente seleziona e consolida quali esperienze meritano di essere trasferite alla memoria a lungo termine.

La quarta convergenza riguarda il ruolo della **causalità**. Nel framework AMI, la capacità di modellare relazioni causali è implicita nell'obiettivo JEPA: un predittore che deve anticipare la rappresentazione futura di una scena dati il contesto corrente deve, per fare previsioni accurate, apprendere qualcosa delle leggi che governano l'evoluzione del mondo. Gögl e Yau (2026) rendono esplicita questa connessione mostrando che il prior condizionale appreso $p_\theta(s_\gamma \mid s_x, z)$ nel modello generativo corrispondente a JEPA funge da codificatore di relazioni causali nello spazio latente: la variabile ausiliaria z cattura quella componente della variabilità del target che non è determinata causalmente dal contesto ma è invece stochasticamente indipendente da esso. Mumuni e Mumuni trattano la causalità esplicitamente come uno dei quattro pilastri dell'AGI, distinguendo fra causalità associativa (livello 1 nella gerarchia di Pearl), intervenzionale (livello 2) e controfattuale (livello 3), e sostenendo che i LLM attuali, nonostante la loro impressionante capacità di recuperare correlazioni statistiche dai dati, non hanno acquisito una comprensione genuina delle relazioni causa-effetto.

17.6 Le divergenze irriducibili: dove i due paradigmi si separano

Le convergenze appena descritte, per quanto reali e significative, non devono oscurare le divergenze che rimangono profonde e strutturalmente irriducibili. Esse si situano a tre livelli distinti: il livello del substrato computazionale, il livello della teoria dell'embodiment, e il livello della relazione fra linguaggio e cognizione.

17.6.1 Il substrato computazionale: predizione generativa vs. predizione nell'embedding space

La divergenza più tecnica riguarda la scelta dell'obiettivo di apprendimento e il suo rapporto con la struttura dello spazio delle rappresentazioni. LeCun sostiene con vigore che qualsiasi obiettivo che richieda al modello di ricostruire l'input nella sua forma grezza — sia esso un'immagine, un frammento audio o una sequenza di token — impone al modello di codificare informazioni irrilevanti per la comprensione delle strutture del mondo. Il famoso argomento delle «features non informative per la percezione» (*uninformative features for perception*) che Balestriero e LeCun (2024) hanno formalizzato in modo rigoroso stabilisce che gli autoencoder addestrati con obiettivi di ricostruzione tendono a specializzarsi nella codifica dei dettagli di bassa frequenza dell'input, che risultano facili da predire, a scapito delle strutture semantiche di alto livello che sono computazionalmente più costose da apprendere ma molto più utili per il ragionamento e la pianificazione.

Il risultato sperimentale di LLM-JEPA sull'analisi dello spazio delle rappresentazioni offre un'illustrazione diretta di questa tesi nel dominio del linguaggio. Huang, LeCun e Balestriero mostrano che il termine JEPA nella perdita composta ha un effetto misurabile sulla geometria dello spazio latente: la decomposizione SVD della differenza Enc(Text) − Enc(Code) rivela valori singolari significativamente inferiori per i modelli addestrati con $\mathcal{L}$_LLM-JEPA rispetto ai modelli addestrati con il solo $\mathcal{L}$_LLM. Questa compressione dello spazio singolare indica che la mappa dall'embedding del testo all'embedding del codice è approssimativamente confinata in un sottospazio di dimensione ridotta, e che questa struttura compatta è approssimativamente lineare: il residuo dell'errore di regressione ai minimi quadrati

$$\min_X \|\mathrm{Enc}(\mathrm{Text}) \cdot X - \mathrm{Enc}(\mathrm{Code})\|^2$$

scende da valori dell'ordine di 10^3 per il modello base e per il modello fine-tuned con la sola $\mathcal{L}$_LLM a valori dell'ordine di unità per i modelli fine-tuned con $\mathcal{L}$_LLM-JEPA. Questa quasi-linearità della mappa fra le due viste non è un artificio, bensì una proprietà emergente dell'obiettivo variazionale: minimizzare la distanza coseno fra Pred(Enc(Text)) e Enc(Code) induce una struttura affine nello spazio degli embedding che si manifesta come riduzione della dimensionalità intrinseca della relazione fra le due viste.

La prospettiva di Mumuni e Mumuni su questo punto è radicalmente diversa. Il loro framework non propone di modificare l'obiettivo di pretraining degli LLM — esso rimane la predizione autoregressiva del token successivo — bensì di *arricchire* i Large Language Models con principi cognitivi aggiuntivi che vengono implementati come moduli separati o come obiettivi di fine-tuning. L'architettura che Mumuni e Mumuni propongono nell'ultimo capitolo del loro survey integra embodiment, grounding, causalità e memoria come strati sovrapposti a un core LLM che continua a fare ciò che sa fare meglio: generare testo

coerente e contestualmente appropriato. Questa scelta architetturale riflette una valutazione implicita della necessità di preservare le capacità generative degli LLM come punto di partenza irrinunciabile, e di costruire attorno ad esse le capacità cognitive che mancano.

17.6.2 La teoria dell'embodiment: simulazione vs. radicamento sensorimotorio

La seconda divergenza riguarda il modo in cui i due paradigmi concepiscono il radicamento corporeo dell'intelligenza. Nel framework AMI, l'embodiment è presente ma *virtuale*: il sistema apprende un modello del mondo che codifica le relazioni causali fra azioni e stati, ma questo apprendimento avviene prevalentemente in ambienti simulati o a partire da dati osservazionali passivi. LeCun non esclude l'interazione fisica con il mondo, ma il suo framework è progettato in modo che il modello del mondo possa essere pre-addestrato in modo auto-supervisionato su grandi quantità di dati senza richiedere un robot fisicamente incarnato.

Mumuni e Mumuni, per converso, attribuiscono all'embodiment un ruolo più radicale e fondativo: l'intelligenza non è semplicemente *facilitata* dall'esperienza incarnata, ma ne è *costituita*. Essi seguono la tradizione della cognizione incarnata (*embodied cognition*) sviluppata da Wilson e Golonka (2013) e da altri ricercatori nel campo delle scienze cognitive, secondo cui i processi cognitivi del sistema nervoso umano sono profondamente radicati nelle interazioni del cervello con il corpo e con l'ambiente esterno. In questa prospettiva, il cervello, il corpo e l'ambiente non sono tre componenti separate di un sistema in cui il primo governa le altre due, ma formano un sistema unitario che si co-specifica reciprocamente attraverso l'interazione continua.

Le implicazioni di questa divergenza per la progettazione di sistemi AGI sono significative. Se si adotta la prospettiva AMI, è in linea di principio possibile addestrare un modello del mondo su dati raccolti passivamente — video di sorveglianza, sequenze di immagini, registrazioni audio — e poi trasferirlo su un agente robotico che utilizzerà il modello pre-addestrato per pianificare le proprie azioni. Se si adotta la prospettiva di Mumuni e Mumuni, questo trasferimento è possibile ma non sufficiente: le rappresentazioni apprese da dati passivi catturano strutture statistiche ma non le strutture causali e sensorimotorie che emergono dall'interazione *deliberata* con il mondo fisico — cioè dall'esperienza in prima persona di come le proprie azioni modificano l'ambiente e producono feedback sensoriali. La distinzione cruciale, che Mumuni e Mumuni sottolineano con forza, è quella fra osservazione passiva e esplorazione attiva: soltanto nel secondo caso l'agente «controlla i dati che riceve attraverso le interazioni con l'ambiente» e può quindi apprendere le strutture causali in modo genuino piuttosto che per correlazione statistica.

17.6.3 La relazione fra linguaggio e cognizione: strumento vs. fondamento

La terza e forse più profonda divergenza riguarda il ruolo che il linguaggio naturale svolge nell'economia della cognizione intelligente. LeCun è esplicito nel ritenere che i Large Language Models autoregressivi non possano costituire il fondamento dell'AGI, per ragioni che vanno ben al di là della semplice critica all'obiettivo di pretraining. Il linguaggio, nella prospettiva AMI, è un **mezzo di comunicazione** che permette di trasmettere conoscenza e

di coordinare l'azione fra agenti, ma non è la struttura portante della cognizione. L'intelligenza — sia biologica sia artificiale — deve radicarsi in un'esperienza del mondo che precede e sopravanza il linguaggio: la comprensione delle proprietà fisiche degli oggetti, delle relazioni spaziali e temporali, dei vincoli causali che governano l'evoluzione degli eventi. Questa comprensione si acquisisce attraverso l'interazione sensorimotoria con il mondo, non attraverso la lettura di testi.

Mumuni e Mumuni assumono una posizione diametralmente opposta su questo punto, ispirandosi a una tradizione di ricerca in linguistica cognitiva — rappresentata in particolare dai lavori di Lupyan (2016) e di Clark (2012) — che attribuisce al linguaggio un ruolo costitutivo nella cognizione piuttosto che meramente strumentale. Essi citano l'evidenza empirica secondo cui le differenze linguistiche influenzano il modo in cui gli individui percepiscono e ricordano le esperienze, e sostengono che il linguaggio non è soltanto un sistema statico di simboli che denotano oggetti ed eventi del mondo reale, ma uno **strumento per manipolare simboli** e per costruire inferenze di ordine superiore. In questa prospettiva, la straordinaria versatilità dei Large Language Models — la loro capacità di ragionare su testi diversissimi, di rispondere a domande in zero-shot, di generare codice corretto e prose creative — non è un artefatto superficiale ma il prodotto di una forma genuina di «conoscenza augmentata dal linguaggio», per usare l'espressione di Lupyan.

La difficoltà di questa divergenza è che, al momento attuale, entrambe le posizioni sono empiricamente sostenibili. I successi dei Large Language Models su compiti che sembravano richiedere comprensione profonda — ragionamento matematico, pianificazione logistica, comprensione del linguaggio in contesti ambigui — suggeriscono che i dati linguistici codificano più struttura del mondo di quanto la prospettiva AMI sembri concedere. Al tempo stesso, i fallimenti sistematici degli LLM su compiti di ragionamento causale — documentati da Zečević et al. (2023) e richiamati da Mumuni e Mumuni — confermano che l'apprendimento statistico di pattern linguistici non equivale all'acquisizione di una comprensione genuina delle relazioni causa-effetto, che richiede strutture di rappresentazione diverse da quelle implementate dai Transformer autoregressivi.

17.7 La sintesi parziale di LLM-JEPA e Var-JEPA: verso un avvicinamento?

I due paper tecnici qui esaminati suggeriscono che le divergenze descritte nella sezione precedente non sono necessariamente insuperabili, e che le linee di ricerca più promettenti si situano precisamente nell'intersezione fra i due paradigmi.

LLM-JEPA dimostra empiricamente che un obiettivo JEPA — nato nel contesto del framework AMI per la visione — può essere applicato con successo ai Large Language Models senza sacrificarne le capacità generative. Il risultato teorico più rilevante, già discusso in termini tecnici ma che merita di essere ripreso qui in chiave interpretativa, è che la minimizzazione di $\mathcal{L}$_LLM-JEPA *non* minimizza implicitamente $\mathcal{L}$_LLM, e viceversa: i due termini della perdita composta sono genuinamente complementari e non ridondanti. Questo risultato è cruciale perché smentisce la tesi — che una lettura superficiale delle posizioni di LeCun potrebbe indurre — secondo cui l'obiettivo JEPA e l'obiettivo autoregressivo siano incompatibili o che il secondo debba essere abbandonato per far posto al primo. La funzione

di perdita composita $\mathcal{L}$_LLM-JEPA suggerisce invece che i due obiettivi sono *ortogonali* nello spazio delle funzioni apprese: il termine LLM favorisce la capacità generativa, il termine JEPA favorisce la struttura astratta dello spazio latente, e i due effetti si sommano senza cancellarsi.

Var-JEPA porta questa riconciliazione a un livello ulteriore, mostrando che il JEPA canonico — con la sua dipendenza da artifici architetturali come stop-gradient e EMA — è interpretabile come un caso limite deterministica di un modello generativo probabilistico. La conseguenza più immediata di questa reinterpretazione è che la dicotomia «JEPA vs. modelli generativi» — su cui LeCun ha costruito parte della propria argomentazione programmatica — perde la propria radicalità: JEPA e modelli latenti variazionali non sono due classi di approcci opposte ma due punti nello stesso spazio progettuale, separati dalla presenza o assenza di una struttura probabilistica esplicita sulle variabili latenti.

La corrispondenza matematica che Gögl e Yau stabiliscono può essere resa ancora più precisa nei termini del quinto termine dell'ELBO. Il termine:

$$-\mathrm{KL}(q_\phi(s_\gamma \mid s_x, z, y) \,\|\, p_\theta(s_\gamma \mid s_x, z))$$

decompone, attraverso la chirurgia dell'ELBO (*ELBO surgery*) introdotta da Hoffman e Johnson (2016), in una divergenza fra la distribuzione posteriore aggregata del target e il prior condizionale, e un termine di bottleneck informativo:

$$\mathbb{E}_y[\mathrm{KL}(q_\phi(s_\gamma \mid s_x, z, y) \,\|\, p_\theta(s_\gamma \mid s_x, z))] \geq \mathrm{KL}(q_\phi(s_\gamma) \,\|\, p_\theta(s_\gamma \mid s_x, z)) + I_\{q_\phi\}(y;\, s_\gamma \mid s_x,\, z)$$

dove $I_\{q_\phi\}(y;\, s_\gamma \mid s_x,\, z)$ è l'informazione mutua fra l'osservazione target y e la sua rappresentazione latente s_γ, condizionata sulle variabili di contesto. Questo termine di informazione mutua cattura precisamente l'entità in cui il codificatore del target utilizza l'osservazione effettiva y per aggiustare la rappresentazione latente oltre a ciò che il predittore contestuale già anticipa. Nella versione deterministica di JEPA, questo termine è implicitamente massimizzato — il codificatore target deve necessariamente basarsi sull'osservazione effettiva y per produrre la propria rappresentazione — ma non è regolarizzato: nulla garantisce che la rappresentazione target sia strutturata in modo da essere predittibile dal contesto. Nel Var-JEPA, al contrario, la pressione dei termini KL verso il prior condizionale appreso $p_\theta(s_\gamma \mid s_x, z)$ garantisce che le rappresentazioni target siano tanto informative per il task downstream quanto predittibili a partire dal contesto.

Questa struttura teorica ha una risonanza diretta con il problema del *symbol grounding* che Mumuni e Mumuni pongono al centro della propria analisi. Il fondamento simbolico — la capacità di connettere rappresentazioni astratte interne a entità e concetti del mondo reale — richiede che le rappresentazioni latenti apprese dal sistema siano *semanticamente ancorate*, ovvero che catturino le strutture del mondo che sono causalmente rilevanti piuttosto che i pattern statistici superficiali presenti nei dati di addestramento. L'obiettivo Var-JEPA, attraverso la pressione combinata del termine di ricostruzione e dei termini KL, produce rappresentazioni che soddisfano questa condizione in modo più robusto rispetto al JEPA canonico, come confermano i risultati sperimentali di Gögl e Yau sulla preservazione dell'incertezza latente nei dataset tabulari.

17.8 Il confronto sul piano della quantificazione dell'incertezza

Un aspetto del confronto che merita attenzione specifica, e che non emerge in modo diretto dalla letteratura sulla cognizione incarnata discussa da Mumuni e Mumuni, è la capacità dei diversi framework di quantificare l'incertezza nelle proprie previsioni. Nella prospettiva AMI classica, il world model JEPA è deterministica: dato un contesto x e una variabile ausiliaria z, il predittore produce un'unica stima puntuale $\hat{s}_\gamma$ della rappresentazione target. L'incertezza è gestita implicitamente attraverso la variabilità dell'ensemble di valori z campionati durante la pianificazione, ma non è quantificata in modo principiato come distribuzione di probabilità sullo spazio delle rappresentazioni.

Il Var-JEPA risolve questo problema in modo elegante, fornendo stime di incertezza per campione (*per-sample uncertainty estimates*) attraverso la varianza della distribuzione posteriore delle variabili latenti. Per un'osservazione target y e un contesto x, l'incertezza latente è data dalla varianza della distribuzione posteriore $q_\phi(s_\gamma | s_x,z,y)$, che Gögl e Yau aggregano per ottenere uno scalare di incertezza per campione. I risultati sul dataset MNIST corrotto e sul dataset sintetico SIM mostrano che questa stima di incertezza è correlata in modo significativo con il grado di corruzione o ambiguità dell'osservazione: il coefficiente di Spearman fra l'incertezza latente standardizzata e il punteggio di incertezza simulato è 0.949 per MNIST e 0.640 per SIM, valori che attestano una correlazione molto elevata nel primo caso e moderata nel secondo.

Questa capacità di quantificare l'incertezza ha implicazioni dirette per il problema della *self-awareness* discusso da Mumuni e Mumuni. Un sistema AGI self-aware deve essere in grado di riconoscere quando la propria conoscenza interna è insufficiente per affrontare un problema e di ricorrere di conseguenza a risorse esterne — un meccanismo analogo a quello degli esseri umani quando consultano esperti o libri di testo per problemi che eccedono le proprie competenze dirette. La correlazione fra incertezza latente e incertezza effettiva osservata nei risultati di Var-JEPA suggerisce che un sistema addestrato con obiettivi variazionali acquisisce, come proprietà emergente del processo di apprendimento, una forma di metacognizione implicita: sa, per così dire, quello che non sa, nel senso che la struttura probabilistica delle proprie rappresentazioni riflette il grado di ambiguità delle osservazioni che ha incontrato durante l'addestramento.

17.9 Sintesi critica: asimmetrie residue e lacune aperte

La rassegna condotta in questo capitolo permette di delineare un quadro più preciso delle asimmetrie che persistono fra il framework AMI e l'approccio LLM-centrico di Mumuni e Mumuni, dopo aver identificato le convergenze e i punti di avvicinamento descritti nelle sezioni precedenti.

La prima asimmetria riguarda la **scalabilità** degli approcci. LLM-JEPA e Var-JEPA sono stati validati, nei paper qui discussi, su modelli relativamente piccoli (da 1B a 8B parametri) e su dataset specializzati. La questione di come l'obiettivo JEPA si comporta a scala di pretraining

completo — su centinaia di miliardi di token e modelli con decine di miliardi di parametri — rimane aperta. Gli autori di LLM-JEPA riconoscono esplicitamente questo limite e ne indicano la risoluzione come una direzione prioritaria di ricerca futura. Dal canto loro, i sistemi LLM discussi da Mumuni e Mumuni includono modelli all'avanguardia come GPT-4, PaLM-E e LLaVA, il cui addestramento è avvenuto a una scala che non è ancora stata raggiunta dagli obiettivi JEPA per il linguaggio.

La seconda asimmetria riguarda la **generalizzazione cross-dominio**. Il framework AMI è progettato come un'architettura universale che prescinde dal dominio sensoriale: la stessa struttura modulare con world model JEPA, configuratore, costi e attore può essere istanziata per la visione, per il linguaggio, per la robotica sensorimotoria. Mumuni e Mumuni propongono un framework anch'esso universale nei principi — embodiment, grounding, causalità, memoria come principi agnostici rispetto all'architettura — ma le soluzioni concrete che discutono sono prevalentemente incentrate sul dominio linguistico e visivo-linguistico, con incursioni nel dominio della robotica incarnata. La tensione fra universalità dei principi e specificità delle implementazioni è comune a entrambi gli approcci.

La terza asimmetria, forse la più fondamentale, riguarda il **trattamento del ragionamento causale**. Il framework AMI propone di apprendere strutture causali attraverso l'interazione del modello del mondo con ambienti simulati fisicamente plausibili — il che presuppone che le leggi della fisica newtoniana e della termodinamica siano già incorporate nella simulazione, garantendo che l'agente apprenda causas reali anziché spurie correlazioni. Il framework di Mumuni e Mumuni propone invece di integrare la causalità nei LLM attraverso una combinazione di modelli causali grafici, modelli di equazioni strutturali (Structural Causal Models, SCM) e grafi di conoscenza, che codificano esplicitamente le relazioni causa-effetto come struttura a priori del sistema. Entrambi gli approcci evitano di affidarsi alla sola statistica associativa come sostituto dell'inferenza causale — e in questo senso convergono nella diagnosi — ma propongono soluzioni architetturalmente incompatibili.

Nessuno dei due paper tecnici qui esaminati — LLM-JEPA e Var-JEPA — affronta direttamente il problema del ragionamento causale. LLM-JEPA dimostra che l'obiettivo JEPA migliora le prestazioni su task di ragionamento matematico (GSM8K) e di parsing sintattico (Spider), ma non fornisce evidenza diretta che questo miglioramento rifletta un'acquisizione di capacità causali piuttosto che un miglioramento nella qualità delle rappresentazioni semantiche. Var-JEPA, attraverso la variabile ausiliaria z e il suo prior condizionale appreso $p_\theta(s_\gamma | s_x, z)$, offre un meccanismo latente per catturare la variabilità causalmente irriducibile del target rispetto al contesto, ma la sua validazione empirica si concentra sui dati tabulari e non include task specifici di ragionamento causale.

Queste lacune aperte non infirmano la portata delle convergenze identificate. Esse indicano piuttosto i fronti su cui la ricerca futura dovrà avanzare per costruire una sintesi davvero coerente fra le intuizioni del framework AMI e quelle dell'approccio LLM-centrico: un'architettura che combini la struttura predittiva in spazio latente di JEPA con i principi cognitivi di embodiment, grounding, causalità e memoria che Mumuni e Mumuni identificano come irrinunciabili per l'AGI.

Capitolo 18 — JEPA Probabilistico, LLM-JEPA e AGI con LLM: un Quadro Unificato?

18.1 La questione dell'unificazione e i suoi limiti metodologici

Prima di avventurarsi nella costruzione di un quadro unificato che tenti di connettere le tre linee di ricerca analizzate nei capitoli precedenti — il JEPA probabilistico di Gögl e Yau, l'estensione ai Large Language Models di Huang, LeCun e Balestriero, e il framework fondazionale per l'AGI di Mumuni e Mumuni — è opportuno mettere a fuoco con precisione che cosa si intende per «unificazione» in questo contesto, e quali sono i confini epistemologici entro cui una tale operazione intellettuale è legittima.

L'unificazione di cui si parla non è di tipo riduzionistico: non si afferma che uno dei tre approcci sia un caso speciale degli altri due, né che esista un singolo formalismo che li contenga tutti come istanziazioni particolari. Si tratta piuttosto di un'unificazione di tipo **strutturale e corrispondenziale**: si mostra che le strutture matematiche, le astrazioni architetturali e i principi cognitivi che i tre approcci propongono si sovrappongono in modo non banale, si completano vicendevolmente nelle zone in cui ciascuno è parziale o silenzioso, e insieme delineano un'architettura cognitiva più ricca di quanto ciascuno di essi singolarmente suggerisca. Questa operazione richiede un passaggio non banale: occorre tradurre le formulazioni tecniche dei paper in un linguaggio astratto sufficiente a rivelare le corrispondenze, senza perdere la specificità matematica che le rende verificabili e falsificabili.

Un secondo avvertimento metodologico riguarda il grado di speculazione implicito nel tentativo di costruire tale quadro. I tre paper da cui si parte sono pubblicazioni recenti — due preprint e un paper in corso di revisione — e i risultati empirici che presentano, per quanto significativi, riguardano scale e domini di applicazione limitati. Qualsiasi proposta di sintesi che proietti queste architetture verso un sistema AGI completo va trattata come un'ipotesi di ricerca da verificare, non come una conclusione stabilita. Il capitolo è scritto in questo spirito: con rigore nelle deduzioni formali e cautela nelle affermazioni prognostiche.

18.2 Le strutture matematiche dei tre approcci: una mappatura preliminare

Il punto di partenza per qualsiasi tentativo di unificazione è la messa in corrispondenza delle strutture matematiche fondamentali dei tre approcci. Questa operazione rivela che, a un livello di astrazione appropriato, i tre framework condividono una grammatica formale comune incentrata su quattro elementi: **spazi di rappresentazione latente**, **processi di inferenza** (che mappano le osservazioni nelle rappresentazioni), **processi generativi** (che mappano le rappresentazioni nelle osservazioni), e **obiettivi di apprendimento** (che bilanciano fedeltà predittiva e regolarizzazione distribuzionale).

Nel Var-JEPA di Gögl e Yau, questi quattro elementi si articolano come segue. Gli spazi di rappresentazione latente sono tre: lo spazio del contesto ($s_x \in \mathbb{R}^d$), lo spazio ausiliario ($z \in \mathbb{R}^{dz}$) e lo spazio del target ($s_\gamma \in \mathbb{R}^d$). I processi di inferenza sono le distribuzioni posteriori variazionali:

$q_\phi(s_x \mid x) = \mathcal{N}(s_x;\ \mu\hat{}s_x_\phi(x),\ \Sigma\hat{}s_x_\phi(x))$
$q_\phi(z \mid s_x) = \mathcal{N}(z;\ \mu\hat{}z_\phi(s_x),\ \Sigma\hat{}z_\phi(s_x))$
$q_\phi(s_\gamma \mid s_x,z,y) = \mathcal{N}(s_\gamma;\ \mu\hat{}s_\gamma_\phi(s_x,z,y),\ \Sigma\hat{}s_\gamma_\phi(s_x,z,y))$

I processi generativi sono i decoder e il prior condizionale:

$p_\theta(x \mid s_x) = \mathcal{N}(x;\ U\hat{}x_\theta(s_x),\ \sigma^2_x\, I)$
$p_\theta(y \mid s_\gamma) = \mathcal{N}(y;\ U\hat{}y_\theta(s_\gamma),\ \sigma^2_y\, I)$
$p_\theta(s_\gamma \mid s_x,z) = \mathcal{N}(s_\gamma;\ \mu\hat{}s_\gamma_\theta(s_x,z),\ \Sigma\hat{}s_\gamma_\theta(s_x,z))$

L'obiettivo di apprendimento è l'ELBO a cinque termini già derivato nel capitolo precedente, che nella sua forma di perdita pesata prende la struttura:

$\mathcal{L}_ELBO = \alpha_rec \cdot \mathcal{L}_rec + \alpha_gen \cdot \mathcal{L}_gen + \alpha\hat{}KL_s_x \cdot \mathcal{L}\hat{}KL_s_x + \alpha\hat{}KL_z \cdot \mathcal{L}\hat{}KL_z + \alpha\hat{}KL_s_\gamma \cdot \mathcal{L}\hat{}KL_s_\gamma$

Nel framework LLM-JEPA di Huang, LeCun e Balestriero, la stessa grammatica si declina in modo diverso ma strutturalmente analogo. Lo spazio di rappresentazione latente è unico — lo spazio degli embedding del Transformer — e le due viste (testo e codice, o contesto e completamento) sono proiettate in questo spazio attraverso lo stesso codificatore, ma con maschere di attenzione distinte che impediscono il flusso informativo tra le viste durante la codifica. Il processo di inferenza è il codificatore del Transformer stesso, che produce:

$Enc(Text) \equiv s_x, \quad Enc(Code) \equiv s_\gamma$

Il processo generativo è duplice: la testa autoregressiva del Transformer implementa la generazione token per token del codice a partire dal testo (che corrisponde al decoder $p_\theta(y \mid s_\gamma)$ nel Var-JEPA), mentre il predittore JEPA — implementato attraverso l'aggiunta di token [PRED] alla fine della sequenza di testo — implementa la predizione della rappresentazione latente del codice a partire dalla rappresentazione latente del testo (che corrisponde al prior condizionale appreso $p_\theta(s_\gamma \mid s_x,z)$ nel Var-JEPA, con $z = 0$ implicitamente poiché LLM-JEPA non dispone di una variabile ausiliaria esplicita). L'obiettivo è la perdita composita:

$\mathcal{L}_LLM\text{-}JEPA = \sum_\{\ell=2\}\hat{}\{L\}\ \mathcal{L}_LLM(Text_\{1:\ell\text{-}1\},\ Text_\ell) + \lambda \cdot d(Pred(Enc(Text)),\ Enc(Code))$

dove $d(\cdot,\cdot)$ è la dissimilarità coseno, che corrisponde formalmente a $-\cos(Pred(Enc(Text)),\ Enc(Code)) = -(Pred(Enc(Text)) \cdot Enc(Code)) \ /\ (\|Pred(Enc(Text))\|_2 \cdot \|Enc(Code)\|_2)$.

Nel framework di Mumuni e Mumuni, la grammatica formale è meno esplicita sul piano matematico — il survey opera prevalentemente a livello architetturale e cognitivo — ma è possibile identificare corrispondenze precise con i quattro elementi della grammatica comune. Gli spazi di rappresentazione latente corrispondono alle diverse modalità di memoria identificate da Mumuni e Mumuni: la memoria semantica (conoscenza fattuale e relazionale di lungo termine), la memoria episodica (esperienze specifiche con contesto temporale e spaziale), la memoria procedurale (sequenze di azioni per compiti sensorimotori) e la memoria di lavoro (informazioni attive per il compito corrente). I processi di inferenza corrispondono ai moduli percettivi — encoders multimodali che trasformano input sensoriali grezzi in rappresentazioni latenti — e ai meccanismi di recupero dalla memoria a lungo termine (Retrieval-Augmented Generation, grafi di conoscenza, database vettoriali). I processi generativi corrispondono ai meccanismi di generazione del testo e di pianificazione motoria. Gli obiettivi di apprendimento, nel framework di Mumuni e Mumuni, non sono specificati a livello di singola funzione di perdita ma sono distribuiti su più fasi di addestramento (pretraining autoregressivo, fine-tuning supervisionato, apprendimento per rinforzo con feedback umano).

18.3 Quattro corrispondenze strutturali fondamentali

La mappatura preliminare descritta nella sezione precedente rivela quattro corrispondenze strutturali che meritano di essere discusse con il dettaglio che la loro importanza richiede. Esse non sono mere analogie retoriche ma connessioni formali che si manifestano a livello delle strutture matematiche dei tre approcci.

18.3.1 Embodiment come interfaccia osservazione-rappresentazione

Il concetto di embodiment — il radicamento corporeo dell'intelligenza — che Mumuni e Mumuni pongono al primo posto nella loro gerarchia di principi cognitivi fondamentali corrisponde, nel linguaggio del Var-JEPA, ai **processi generativi osservativi**: i decoder $p_\theta(x \mid s_x)$ e $p_\theta(y \mid s_y)$ che mappano dalle rappresentazioni latenti alle osservazioni crude, e i loro inversi funzionali — i processi di inferenza $q_\phi(s_x \mid x)$ e $q_\phi(s_y \mid s_x,z,y)$ — che mappano dalle osservazioni alle rappresentazioni.

Questa corrispondenza non è banale. Nel Var-JEPA, i termini di ricostruzione $\mathcal{L}_rec = -\mathbb{E}_\{q_\phi(s_x \mid x)\}[\log p_\theta(x \mid s_x)]$ e $\mathcal{L}_gen = -\mathbb{E}_\{q_\phi(s_x,z,s_y \mid x,y)\}[\log p_\theta(y \mid s_y)]$ hanno il ruolo esplicito di vincolare le rappresentazioni latenti a preservare l'informazione necessaria per ricostruire le osservazioni originali. Questo vincolo è precisamente ciò che Mumuni e Mumuni intendono quando affermano che il radicamento corporeo «fornisce la struttura e i meccanismi per interfacciarsi e interagire con il mondo»: le rappresentazioni latenti non sono entità astratte fluttuanti in uno spazio algebrico senza ancoraggio semantico, ma codici compressi dell'esperienza osservativa che mantengono una connessione funzionale con il mondo fisico attraverso la relazione bidirezionale codificatore-decodificatore.

Nel caso tabellare trattato da Gögl e Yau, questa connessione è implementata attraverso decoder feature-specifici che ricostruiscono indipendentemente ciascuna variabile osservata:

$$p_\theta(x\hat{}(j) \mid s\hat{}(j)_x) = \mathcal{N}(x\hat{}(j);\, u\hat{}(j)_\theta(s\hat{}(j)_x),\, \sigma^2) \quad \text{per } j \in \mathcal{J}_num$$
$$p_\theta(x\hat{}(j) \mid s\hat{}(j)_x) = \text{Categorical}(x\hat{}(j);\, \text{softmax}(u\hat{}(j)_\theta(s\hat{}(j)_x))) \quad \text{per } j \in \mathcal{J}_cat$$

dove $u^{(j)}_\theta : \mathbb{R}^{Dhid} \to \mathbb{R}^1$ (numeriche) e $u^{(j)}_\theta : \mathbb{R}^{Dhid} \to \mathbb{R}^{cj}$ (categoriche con C categorie) sono reti neurali feature-specifiche. L'estensione al caso multimodale — immagini, audio, testo, segnali propriocettivi — richiede semplicemente di sostituire questi decoder scalari con decoder appropriati per ciascuna modalità sensoriale, mantenendo invariata la struttura dell'ELBO.

L'aspetto più interessante di questa corrispondenza è che il Var-JEPA fornisce una misura quantitativa del grado di embodiment di un sistema: la qualità della ricostruzione, misurata attraverso $\mathcal{L}_{rec}$ e $\mathcal{L}_{gen}$, indica quanto le rappresentazioni latenti preservano delle strutture fisicamente rilevanti dell'esperienza. Un sistema con $\mathcal{L}_{rec}$ elevato ha rappresentazioni latenti che non riescono a catturare adeguatamente le strutture sensoriali, e quindi è, in senso tecnico, «scarsamente incarnato». Un sistema con $\mathcal{L}_{rec}$ basso ha rappresentazioni che mantengono un ancoraggio fisico robusto e può quindi generare simulazioni sensorialmente plausibili dell'esperienza — che è precisamente ciò che LeCun intende per «world model».

18.3.2 Symbol grounding come prior condizionale appreso

Il *symbol grounding* — la connessione fra rappresentazioni astratte interne e referenti nel mondo fisico — corrisponde, nel Var-JEPA, al **prior condizionale appreso** $p_\theta(s_\gamma | s_x, z)$. Questo componente, che nel JEPA canonico corrisponde semplicemente al predittore $g(s_x, z)$, è il cuore dell'architettura: è esso che impara a mappare dalla rappresentazione di un contesto alla distribuzione sulle possibili rappresentazioni del target, senza accedere direttamente all'osservazione target.

La ragione per cui questa corrispondenza è strutturalmente profonda — e non meramente analogica — è che il problema del fondamento simbolico, nella sua formulazione classica di Harnad (1990), richiede che i simboli interni di un sistema acquisiscano significato attraverso la loro connessione con le entità del mondo fisico che denotano. Il prior condizionale appreso $p_\theta(s_\gamma | s_x, z)$ implementa precisamente questo tipo di connessione: esso apprende a predire la rappresentazione latente del target (il «simbolo interno» nel senso di Harnad) a partire dalla rappresentazione del contesto (che è ancorata all'osservazione fisica attraverso il processo di inferenza $q_\phi(s_x | x)$ e il processo generativo $p_\theta(x | s_x)$). Il prior condizionale è quindi il meccanismo attraverso cui i simboli interni — le rappresentazioni latenti — acquisiscono una semantica strutturata, ovvero la capacità di essere predetti e manipolati in relazione ad altri simboli in modo che rispecchi la struttura causale del mondo.

Nel contesto di LLM-JEPA, il predittore Pred(Enc(Text)) gioca lo stesso ruolo: impara a produrre, a partire dalla rappresentazione latente di una descrizione linguistica, una stima della rappresentazione latente del programma o del codice che implementa la funzionalità descritta. Questo processo è, in senso tecnico, un processo di fondamento simbolico: il testo (il «simbolo») viene collegato al codice (l'«entità» che implementa una funzione computabile nel mondo) attraverso un'operazione di predizione nello spazio delle rappresentazioni.

La differenza rispetto al Var-JEPA è che in LLM-JEPA non esiste una variabile ausiliaria z esplicita, e il processo di inferenza non è variazionale ma deterministico. Questo significa che il fondamento simbolico realizzato da LLM-JEPA è meno robusto dal punto di vista della quantificazione dell'incertezza: non è possibile, nel framework LLM-JEPA attuale, distinguere

fra situazioni in cui la relazione contesto-target è altamente determinata e situazioni in cui è ambigua. L'introduzione di una variabile ausiliaria latente — sulla falsariga del Var-JEPA — risolverebbe questo problema e permetterebbe al sistema di esprimere incertezza su quale dei possibili programmi implementi correttamente la funzionalità descritta dal testo.

18.3.3 Causalità come struttura del grafo aciclico diretto

Il terzo principio fondazionale di Mumuni e Mumuni — la causalità — corrisponde, nel Var-JEPA, alla **struttura del grafo aciclico diretto (DAG)** che governa il processo generativo e la sua fattorizzazione:

$$p_\theta(x, y, s_x, z, s_\gamma) = p(s_x) \cdot p(z) \cdot p_\theta(x \mid s_x) \cdot p_\theta(s_\gamma \mid s_x, z) \cdot p_\theta(y \mid s_\gamma)$$

Questo DAG non è un'arbitraria scelta di fattorizzazione matematica: esso codifica esplicitamente relazioni di dipendenza e indipendenza condizionale che corrispondono ad assunzioni causali sulla struttura del mondo. In particolare, il DAG codifica le seguenti indipendenze condizionali:

$$x \perp\!\!\!\perp y \mid s_x, z, s_\gamma$$
$$x \perp\!\!\!\perp z \mid s_x$$
$$x \perp\!\!\!\perp s_\gamma \mid s_x, z$$
$$y \perp\!\!\!\perp s_x, z \mid s_\gamma$$
$$z \perp\!\!\!\perp x \mid s_x$$

La prima indipendenza stabilisce che l'osservazione contestuale x e l'osservazione target y sono condizionalmente indipendenti date le rispettive rappresentazioni latenti e la variabile ausiliaria: tutta l'informazione rilevante dell'una per l'altra è mediata dalle rappresentazioni latenti. La seconda indipendenza stabilisce che la variabile ausiliaria z — che cattura la componente stocastica della relazione contesto-target — è condizionalmente indipendente dall'osservazione grezza x dato il contesto latente s_x: questo garantisce che z non codifichi dettagli sensoriali irrilevanti ma soltanto la struttura astratta della variabilità del target che non può essere inferita dal contesto.

Queste relazioni di indipendenza condizionale corrispondono, nella terminologia di Pearl, a strutture di causalità di livello 1 (associativo) e livello 2 (interventivo). La variabile ausiliaria z può essere interpretata come il «noise» esogeno nella terminologia degli Structural Causal Models (SCM): essa rappresenta i fattori che influenzano il target ma sono causalmente indipendenti dal contesto. La distribuzione $p_\theta(s_\gamma \mid s_x, z)$ è quindi, formalmente, la funzione strutturale che mappa cause (s_x, z) a effetti (s_γ) nel modello causale latente.

Mumuni e Mumuni, nel loro trattamento della causalità, distinguono fra tre approcci principali: i metodi di deep learning puro (che apprendono correlazioni statistiche ma non strutture causali genuine), i metodi neuro-simbolici (che integrano modelli causali grafici come gli SCM nel processo di apprendimento), e i modelli del mondo informati dalla fisica (che utilizzano simulatori fisicamente plausibili per garantire che le strutture causali apprese corrispondano alle leggi reali del mondo). Il Var-JEPA si colloca in una posizione intermedia fra il primo e il secondo approccio: il DAG del processo generativo è specificato a priori

come struttura causale qualitativa, mentre le funzioni che implementano le dipendenze causali (i decoder, il prior condizionale) sono apprese dai dati. Questo è precisamente il regime che Mumuni e Mumuni descrivono come apprendimento causale implicito con struttura esplicita.

18.3.4 Memoria come struttura temporale dell'inferenza

Il quarto principio di Mumuni e Mumuni — la memoria — corrisponde, nel Var-JEPA, al **ruolo che la rappresentazione contestuale s_x svolge come stato interno dell'agente** che persiste attraverso l'episodio corrente e informa la predizione dei target futuri. Questa corrispondenza è però la meno precisa delle quattro, perché il Var-JEPA nella sua formulazione standard è statico: esso opera su coppie (x, y) indipendenti e non prevede meccanismi espliciti per l'accumulo di informazioni attraverso una sequenza temporale di osservazioni.

L'estensione del Var-JEPA al regime dinamico — che richiederebbe di modellare sequenze di osservazioni $(x_1, y_1), (x_2, y_2), ..., (x , y)$ con dipendenze temporali — è indicata da Gögl e Yau come una direzione di ricerca futura, ma non è sviluppata nel paper attuale. La struttura dell'ELBO che essi derivano si presta naturalmente a tale estensione: la rappresentazione contestuale s_x, al tempo t potrebbe essere condizionata non soltanto sull'osservazione corrente x ma anche sulle rappresentazioni contestuali precedenti $\{s_{x,\tau}\}_{\tau<t}$, realizzando così una forma di memoria di lavoro differenziabile.

In termini formali, la distribuzione posteriore del contesto diventerebbe:

$$q_\phi(s_x, \mid x_1,...,x) = q_\phi(s_x, \mid x , h_{t-1})$$

dove h_{t-1} è lo stato nascosto di un encoder ricorrente — per esempio un LSTM o un Transformer con attenzione causale — che riassume l'informazione delle osservazioni precedenti. La distinzione fra memoria di lavoro (h_{t-1}), memoria semantica (il prior $p(s_x)$ che codifica la conoscenza strutturale del dominio) e memoria episodica (le rappresentazioni specifiche $s_{x,\tau}$ per $\tau < t$, recuperate attraverso un meccanismo di attenzione) emerge naturalmente in questa estensione dinamica.

Nel contesto dei Large Language Models, questa struttura è già parzialmente implementata attraverso la finestra di contesto dell'architettura Transformer, che funge da memoria di lavoro, e attraverso i meccanismi di RAG (Retrieval-Augmented Generation) che Mumuni e Mumuni discutono come implementazione della memoria semantica e episodica a lungo termine. LLM-JEPA non affronta esplicitamente la dimensione temporale della memoria, concentrandosi invece sull'apprendimento di rappresentazioni strutturate per singole coppie (testo, codice). Tuttavia, la struttura compatta dello spazio latente indotta dall'obiettivo JEPA — la quasi-linearità della mappa Enc(Text) → Enc(Code) documentata nella Tabella 14 del paper — suggerisce che le rappresentazioni apprese potrebbero facilitare il recupero dalla memoria a lungo termine: una rappresentazione più strutturata è più facilmente interrogabile attraverso meccanismi di retrieval basati sulla similarità nell'embedding space.

18.4 Verso un obiettivo unificato: la perdita JEPA-VAE-LLM

Le quattro corrispondenze strutturali descritte nella sezione precedente permettono di abbozzare la forma generale di un obiettivo di apprendimento unificato che integri, almeno a livello formale, i contributi dei tre framework. Questo obiettivo — che si propone qui come ipotesi di ricerca e non come risultato stabilito — combina la struttura variazionale del Var-JEPA, l'obiettivo generativo autoregressivo di LLM-JEPA e i vincoli cognitivi del framework di Mumuni e Mumuni.

Il punto di partenza è il riconoscimento che l'ELBO del Var-JEPA e la perdita composita di LLM-JEPA possono essere interpretati come casi speciali di un obiettivo più generale definito su un processo generativo multimodale. Consideriamo un sistema che processa m modalità sensoriali $\{x_1, ..., x\}$ — testo, immagini, audio, segnali propriocettivi — e deve apprendere rappresentazioni latenti che siano (a) informativamente compresse rispetto alle osservazioni grezze, (b) predittive delle relazioni fra modalità diverse, (c) strutturate causalmente, e (d) calibrate nell'incertezza.

La distribuzione congiunta del processo generativo si fattorizza come:

$$p_\theta(x_1,...,x , s_1,...,s , z) = p(z) \cdot \prod_i [p(s_i) \cdot p_\theta(x_i | s_i)] \cdot \prod_\{i \neq j\} p_\theta(s | s_i,z)$$

dove s_i è la rappresentazione latente della modalità i, z è la variabile ausiliaria condivisa che cattura la variabilità cross-modale non determinata da nessuna singola modalità, e $p_\theta(s | s_i,z)$ è il prior condizionale che implementa la predizione cross-modale — cioè il predittore JEPA nel suo ruolo di modello del mondo.

La distribuzione posteriore variazionale corrispondente si fattorizza come:

$$q_\phi(s_1,...,s ,z | x_1,...,x) = q_\phi(z | s_1,...,s) \cdot \prod_i q_\phi(s_i | x_i)$$

dove si assume per semplicità che ciascuna rappresentazione modale s_i dipenda soltanto dall'osservazione corrispondente x_i (indipendenza condizionale fra le modalità dato z), e che z dipenda dalla concatenazione di tutte le rappresentazioni modali. Questo è un'ipotesi semplificante che potrebbe essere rilassata in implementazioni più sofisticate.

L'ELBO associato a questo processo generativo si decompone in termini che corrispondono precisamente ai quattro principi di Mumuni e Mumuni:

$$\begin{aligned}
\text{ELBO} = {} & \sum_i \mathbb{E}_\{q_\phi(s_i | x_i)\}[\log p_\theta(x_i | s_i)] && \text{(Embodiment: ricostruzione per modalità)} \\
& + \sum_\{i \neq j\} \mathbb{E}_\{q_\phi\}[\log p_\theta(s | s_i,z)] && \text{(Grounding: predizione cross-modale)} \\
& - \sum_i \mathrm{KL}(q_\phi(s_i | x_i) \,\|\, p(s_i)) && \text{(Struttura causale: regolarizzazione modale)} \\
& - \mathrm{KL}(q_\phi(z | s_1,...,s) \,\|\, p(z)) && \text{(Causalità: regolarizzazione ausiliaria)} \\
& - \sum_\{i \neq j\} \mathrm{KL}(q_\phi(s | x) \,\|\, p_\theta(s | s_i,z)) && \text{(Memoria: compressione predittiva)}
\end{aligned}$$

Il primo termine realizza il radicamento corporeo: le rappresentazioni modali devono essere sufficientemente ricche da permettere la ricostruzione delle osservazioni grezze. Il secondo termine realizza il fondamento simbolico: le rappresentazioni di una modalità devono essere predittive delle rappresentazioni delle altre modalità, attraverso l'operazione del prior condizionale appreso. Il terzo e quarto termine realizzano la struttura causale: le rappresentazioni modali e ausiliarie sono regolarizzate verso prior gaussiani standardizzati, il che impone la struttura causale del DAG e previene il collasso delle rappresentazioni. Il quinto termine realizza la compressione predittiva che è alla base della memoria: le rappresentazioni target vengono spinte a coincidere con le previsioni del prior condizionale, il che equivale a «comprimere» ogni osservazione nello schema causale già appreso.

L'aggiunta del termine autoregressivo di LLM-JEPA — la perdita cross-entropica per la predizione del token successivo nello spazio dell'input linguistico — si integra naturalmente con questa struttura come un termine di regolarizzazione aggiuntivo che preserva le capacità generative nel dominio del linguaggio:

$$\mathcal{L}_{unificato} = -\text{ELBO}_{multimodale} + \gamma \cdot \sum_{\ell} \mathcal{L}_{LLM}(\text{Text}_{1:\ell-1}, \text{Text}_{\ell})$$

dove il parametro $\gamma \geq 0$ bilancia l'importanza relativa delle capacità generative linguistiche rispetto all'apprendimento di rappresentazioni strutturate. Il risultato di Huang, LeCun e Balestriero sul ruolo irrinunciabile di $\mathcal{L}_{LLM}$ per le prestazioni generative — illustrato nella Tabella 10 del paper, dove $\gamma = 0$ porta alla generazione di output vuoti — suggerisce che γ non può essere ridotto a zero senza sacrificare la funzionalità del sistema.

Capitolo 18 — Parte Seconda: L'Architettura Unificata, le Tensioni Interne e i Limiti della Sintesi

18.5 L'architettura integrata: componenti e flussi informativi

La perdita unificata abbozzata nella sezione precedente suggerisce la forma di un'architettura cognitiva che integra i contributi dei tre framework. Prima di descriverla, è necessario premettere che quanto segue è una proposta teorica — un'estrapolazione razionale delle strutture che i tre paper suggeriscono — e non una descrizione di un sistema effettivamente realizzato. I tre paper forniscono i mattoni; la costruzione descritta qui è un'inferenza su come essi potrebbero essere assemblati in modo coerente.

L'architettura proposta si compone di sei livelli funzionali che operano su modalità sensoriali eterogenee e che interagiscono secondo i flussi informativi descritti dal DAG del processo generativo multimodale.

Il **livello di tokenizzazione e codifica modale** trasforma le osservazioni grezze di ciascuna modalità — testo (sequenze di token), immagini (patch visive), segnali propriocettivi (vettori

numerici di posizione e velocità), variabili tabulari (feature numeriche e categoriche) — in una rappresentazione unificata nello spazio degli embedding attraverso encoder specializzati per modalità. Per il testo, l'encoder è il Transformer autoregressivo di LLM-JEPA, che produce la rappresentazione Enc(Text) ≡ s_{ex} attraverso l'embedding dell'ultimo token dell'ultimo layer. Per le immagini, un Vision Transformer (ViT) produce patch embeddings che vengono aggregati attraverso attenzione su un token [CLS]. Per i dati tabulari, la tokenizzazione di Var-T-JEPA — con embedding feature-specifici, codifiche posizionali e processamento Transformer — produce rappresentazioni per ciascuna feature. L'output di questo primo livello è un insieme di distribuzioni posteriori modali:

$$\{q_\phi(s_i \mid x_i) = \mathcal{N}(s_i;\ \mu\hat{}s_i_\phi(x_i),\ \Sigma\hat{}s_i_\phi(x_i))\}_\{i=1\}\hat{}\{m\}$$

Il **livello di aggregazione ausiliaria** inferisce la variabile ausiliaria z che cattura la struttura latente condivisa fra tutte le modalità. Nel Var-JEPA per dati tabulari, questa funzione è svolta dall'auxiliary encoder $q_\phi(z \mid s_x)$ — un MLP compatto che processa la rappresentazione di contesto aggregata attraverso attention pooling. Nell'architettura multimodale qui proposta, questo livello processarebbe la concatenazione o il pool delle rappresentazioni modali $\{s_i\}$ per produrre:

$$q_\phi(z \mid s_1,...,s\ ) = \mathcal{N}(z;\ \mu\hat{}z_\phi(\text{pool}(s_1,...,s\ )),\ \Sigma\hat{}z_\phi(\text{pool}(s_1,...,s\ )))$$

dove pool(·) è un operatore di aggregazione — per esempio la media ponderata attraverso attenzione cross-modale — che produce un vettore a dimensione fissa a partire dalle m rappresentazioni modali.

Il **livello predittivo cross-modale** implementa il prior condizionale appreso $p_\theta(s\ \mid s_i,z)$ per tutte le coppie di modalità (i,j). Questo è il nucleo del sistema — il «world model» nel senso di LeCun — che impara a predire la rappresentazione latente di una modalità a partire da quella di un'altra, attraverso la variabile ausiliaria z che media la variabilità irriducibile. La struttura matematica di questo livello è identica al predittore JEPA: una rete neurale che prende in input (s_i, z) e produce i parametri della distribuzione gaussiana sul target:

$$p_\theta(s\ \mid s_i,z) = \mathcal{N}(s\ ;\ \mu\hat{}s\ _\theta(s_i,z),\ \Sigma\hat{}s\ _\theta(s_i,z))$$

Nel contesto di LLM-JEPA, questo livello corrisponde al predittore Pred(·) applicato alla rappresentazione Enc(Text), con z = 0 implicitamente (assenza di variabile ausiliaria esplicita). L'estensione variazionale richiede l'aggiunta di un campionamento esplicito da $q_\phi(z \mid s_i)$ — la cui struttura è già presente nel Var-JEPA per dati tabulari — e la modifica del predittore per condizionare sia su s_i che su z.

Il **livello decodificativo** ricostruisce le osservazioni grezze a partire dalle rappresentazioni latenti attraverso decoder modalità-specifici. Per il testo, la testa autoregressiva del Transformer implementa $p_\theta(x_text \mid s_text)$ attraverso la distribuzione categorica sui token del vocabolario; per le immagini, un decoder ViT o una rete convoluzionale implementa $p_\theta(x_img \mid s_img)$; per i dati tabulari, i decoder feature-specifici di Var-T-JEPA implementano la distribuzione gaussiana o categorica su ciascuna feature.

Il **livello di memoria** mantiene le rappresentazioni attraverso il tempo e le rende disponibili per l'inferenza corrente attraverso un meccanismo di recupero. Nell'architettura proposta, questo livello si articola in due sottocomponenti: la memoria di lavoro, implementata dalla finestra di contesto del Transformer e dalle rappresentazioni latenti correnti $\{s_i\}$; e la memoria a lungo termine, implementata da un database vettoriale che memorizza le rappresentazioni modali di episodi passati insieme ai loro metadati temporali e contestuali. Il meccanismo di recupero è un'operazione di attenzione cross-episodica che computa la similarità fra la query corrente (la rappresentazione contestuale s_x, al tempo t) e le chiavi memorizzate (le rappresentazioni degli episodi passati), producendo una rappresentazione augmentata:

$$\hat{s}_augmentata = s_x, + \sum_\{\tau<t\} \alpha_\{t,\tau\} \cdot s_x,\tau$$

dove i pesi di attenzione $\alpha_\{t,\tau\} = softmax_\tau(s_x, \cdot s_x,\tau / \sqrt{d})$ sono calcolati attraverso la similarità coseno normalizzata nel latent space.

Il **livello di pianificazione e azione** utilizza le previsioni del mondo model e le stime di costo per pianificare sequenze di azioni ottimali rispetto a un obiettivo specificato dall'agente o dall'utente. Questo livello corrisponde all'attore nel framework AMI, e la sua implementazione concreta è la componente meno specificata nell'architettura proposta: il paper di LLM-JEPA non affronta la pianificazione, il Var-JEPA si limita all'inferenza latente, e Mumuni e Mumuni discutono la pianificazione principalmente come funzione emergente dei LLM avanzati (Chain-of-Thought, EmbodiedGPT) senza proporre un'architettura specifica per integrarla con il framework variazionale.

18.6 Tensioni interne della sintesi proposta

La proposta di unificazione descritta nelle sezioni precedenti presenta tensioni interne significative che è necessario riconoscere esplicitamente, perché esse indicano dove le lacune della sintesi sono più profonde e dove la ricerca futura dovrà concentrare i propri sforzi.

La prima tensione riguarda il **regime di addestramento vs. il regime di inferenza** per la variabile ausiliaria z. Nel Var-JEPA, durante l'addestramento, z è campionata dalla distribuzione posteriore $q_\phi(z|s_x)$ — che ha accesso alla rappresentazione di contesto s_x appresa dall'osservazione grezza x. Durante l'inferenza su nuove osservazioni, z è campionata dal prior standard $p(z) = \mathcal{N}(0, I)$ o dalla distribuzione posteriore $q_\phi(z|s_x)$ se l'osservazione di contesto è disponibile. In LLM-JEPA, non esiste una variabile ausiliaria esplicita: il predittore Pred(Enc(Text)) opera deterministicamente senza campionamento stocastico. Questa asimmetria crea una difficoltà nell'integrazione: un sistema che utilizzi sia l'obiettivo autoregressivo di LLM-JEPA sia l'inferenza variazionale di Var-JEPA deve decidere se e quando campionare da z durante la generazione di testo, e come condizionare la distribuzione del token successivo sia sull'embedding autoregressivo sia sul campione latente z.

Una risoluzione possibile — e questa è una proposta speculativa, non un risultato dei paper — sarebbe trattare z come una variabile di «stile» o «intento» che viene campionata una volta all'inizio di ciascun episodio generativo e poi tenuta fissa durante tutta la generazione della

sequenza. Questo è analogo all'approccio dei modelli linguistici condizionali (conditional language models) in cui un vettore di condizionamento globale influenza la distribuzione del Transformer su tutta la sequenza generata. La differenza rispetto agli approcci standard è che z sarebbe inferita variazionalmente dal contesto — attraverso $q_\phi(z|s_x)$ — piuttosto che specificata esternamente come condizione.

La seconda tensione riguarda il **costo computazionale** dell'obiettivo unificato. LLM-JEPA documenta già un costo aggiuntivo significativo rispetto all'addestramento LLM standard: la necessità di due forward pass aggiuntivi per calcolare Enc(Text) e Enc(Code) con maschere di attenzione separate porta a un raddoppio del costo computazionale per batch. Il meccanismo di *loss dropout* (LD) proposto dagli autori — l'applicazione del termine JEPA soltanto a una frazione α dei batch, con LD rate tipico di 0.5 — riduce il costo incrementale a $(2 - \alpha)$ volte quello standard, ma introduce una nuova iperparametro da tarare. L'estensione al caso multimodale variationale richiederebbe forward pass aggiuntivi per i decoder, per l'encoder ausiliario e per il campionamento attraverso il reparametrization trick, aumentando ulteriormente il costo computazionale. La domanda se questo costo aggiuntivo sia giustificato dai benefici in termini di qualità delle rappresentazioni è empiricamente aperta.

I risultati di Gögl e Yau sul dataset Covertype (CO) gettano una luce interessante su questa questione. Il Var-T-JEPA supera il T-JEPA deterministico in modo consistente — in alcuni casi il T-JEPA collassa completamente (accuratezza 0.408 per MLP+T-JEPA contro 0.679 per MLP+Var-T-JEPA) — ma il confronto non è direttamente con un modello raw-feature. L'overhead computazionale del Var-T-JEPA rispetto al T-JEPA include il campionamento latente, la retropropagazone attraverso il reparametrization trick, e il calcolo dei termini KL in forma chiusa (che per distribuzioni gaussiane diagonali ha una formula analitica che non richiede campionamento):

$$\mathrm{KL}(\mathcal{N}(\mu, \mathrm{diag}(\sigma^2)) \,\|\, \mathcal{N}(0, I)) = \tfrac{1}{2} \sum_{d=1}^{D} (\sigma_d^2 + \mu_d^2 - \log \sigma_d^2 - 1)$$

Questa formula è computazionalmente leggera e differenziabile in modo esatto, il che significa che il principale costo aggiuntivo del Var-JEPA rispetto al JEPA canonico non è nel calcolo dei termini KL ma nella necessità di mantenere distribuzioni di probabilità (media e varianza) per ciascuna variabile latente anziché soltanto un vettore deterministico.

La terza tensione, forse la più fondamentale, riguarda il **problema del collasso parziale** in sistemi multimodali con obiettivi eterogenei. L'analisi di Gögl e Yau mostra che rimuovere i termini KL dalla perdita (variante I della simulazione study) porta a un collasso severo delle distribuzioni latenti, con divergenze KL aggregate nell'ordine di 10-20 e norme medie $\|\mathbb{E}[s]\|_2$ dell'ordine di 3-5. Aggiungere SIGReg ai termini KL rimossi (variante J) riduce parzialmente il collasso ma non lo elimina del tutto: la divergenza KL aggregata scende a circa 2.1 rispetto ai valori dell'ordine di 10.5 della variante I, ma rimane circa 19 volte superiore alla variante con ELBO completo. In un sistema multimodale con $m > 2$ modalità, il rischio di collasso parziale — dove alcune modalità collassano verso costanti mentre altre mantengono struttura — è proporzionalmente più elevato, e la diagnosi e la correzione di tale collasso richiedono procedure di monitoraggio più sofisticate.

Il risultato di Huang, LeCun e Balestriero sull'analisi dei valori singolari di Enc(Text) − Enc(Code) offre un complementare strumento diagnostico: se i primi 100 valori singolari

della differenza fra le rappresentazioni delle due modalità sono compressi in un ristretto sottospazio (come accade nei modelli LLM-JEPA), le due modalità sono allineate nel latent space; se invece i valori singolari sono distribuiti su un range ampio (come nel modello base non fine-tuned), le due modalità occupano regioni distanti e non strutturate dello spazio latente, il che è un segnale di collasso funzionale della predizione cross-modale.

18.7 Il ruolo del loss dropout come ponte fra regimi di addestramento

Il meccanismo di *loss dropout* introdotto da Huang, LeCun e Balestriero (2025) merita un'analisi più approfondita di quella che il paper stesso dedica ad esso, perché esso rivela una proprietà interessante dell'obiettivo JEPA composito che ha implicazioni per il framework unificato proposto in questo capitolo.

Il loss dropout consiste nell'applicare il termine JEPA soltanto a una frazione $(1 - \mathrm{LD})$ dei batch durante l'addestramento, dove $\mathrm{LD} \in [0, 1)$ è il tasso di dropout. Quando il termine JEPA è attivo (per una frazione $1 - \mathrm{LD}$ dei batch), il sistema esegue due forward pass aggiuntivi per ottenere Enc(Text) e Enc(Code) con maschere separate. Quando il termine è disattivato (per una frazione LD dei batch), il sistema esegue soltanto il forward pass standard per l'obiettivo autoregressivo. Il costo computazionale per epoca è quindi:

$$\text{Costo_JEPA-LD} = (1 - \mathrm{LD}) \cdot 3F + \mathrm{LD} \cdot F = (2 - \mathrm{LD}) \cdot F \quad \text{forward pass per batch}$$

dove F è il costo di un singolo forward pass e si assume che ciascuno dei due forward pass aggiuntivi per le due viste costi F.

I risultati empirici mostrano un comportamento controintuitivo: aumentare contemporaneamente LD e λ — mantenendo approssimativamente costante il prodotto $\lambda \cdot (1 - \mathrm{LD})$ — produce prestazioni superiori rispetto all'uso di un valore di LD basso con lo stesso budget computazionale. In termini quantitativi, la variante $\mathrm{LD} = 0.75$, $\lambda = 4$ raggiunge un'accuratezza di 72.42% su NL-RX-SYNTH dopo 6 epoche con 28.98 PFLOPs, contro il 63.57% del LLM-JEPA senza loss dropout ($\mathrm{LD} = 0$, $\lambda = 1$) allo stesso costo computazionale.

Questo risultato può essere interpretato alla luce del concetto di *curriculum di addestramento*: i batch in cui il termine JEPA è attivo agiscono come esempi difficili che forzano il modello a strutturare il proprio spazio latente in modo predittivo cross-modale, mentre i batch in cui il termine è disattivato permettono al modello di consolidare le capacità generative attraverso l'obiettivo autoregressivo standard. La transizione fra i due tipi di batch — determinata stocasticamente dal tasso LD — crea un curriculum implicito che bilancia strutturazione latente e capacità generativa in modo più efficace della semplice combinazione lineare con peso fisso λ.

In termini variazionali, questo comportamento può essere ricondotto alla seguente osservazione: la minimizzazione congiunta di $\mathcal{L}_\mathrm{rec}$ (ricostruzione dell'input) e del termine JEPA (predizione cross-modale) crea una competizione per la capacità rappresentativa delle variabili latenti. La ricostruzione dell'input tende a favorire rappresentazioni ad alta dimensione che codificano tutti i dettagli sensoriali; la predizione cross-modale tende a

favorire rappresentazioni compatte che codificano soltanto la struttura predittiva. Il loss dropout risolve questa competizione non attraverso un compromesso fisso (il peso λ), ma attraverso una *separazione temporale* degli obiettivi: in ciascun batch, il sistema si concentra o sull'uno o sull'altro obiettivo, e l'alternanza fra i due produce, nel lungo periodo, un equilibrio più stabile che quello ottenibile attraverso l'ottimizzazione simultanea.

Questa interpretazione ha un'implicazione diretta per il framework unificato: il parametro γ nella perdita $\mathcal{L}$_unificato = −ELBO_multimodale + γ · $\mathcal{L}$_LLM non dovrebbe essere trattato come un iperparametro fisso ma come una variabile stocastica che alterna i suoi valori fra γ_alto (quando si vuole rafforzare le capacità generative) e 0 (quando si vuole rafforzare la strutturazione latente variazionale). Il loss dropout è quindi un caso speciale del più generale principio di separazione temporale degli obiettivi, che ha un'analoga in molti algoritmi di ottimizzazione stocastica (per esempio la tecnica di annealing ciclico del learning rate).

18.8 La questione della quantificazione dell'incertezza in sistemi unificati

Uno dei contributi più originali del Var-JEPA rispetto al JEPA canonico e a LLM-JEPA è la sua capacità di produrre stime di incertezza per campione a partire dalle varianze delle distribuzioni posteriori latenti. Questa capacità — che Gögl e Yau validano empiricamente sul dataset MNIST corrotto e sul dataset sintetico SIM — è un requisito fondamentale per qualsiasi sistema AGI che aspiri alla *self-awareness* nel senso di Mumuni e Mumuni: un sistema che non sa valutare la propria incertezza non può riconoscere quando la propria conoscenza è insufficiente e non può decidere razionalmente di ricorrere a risorse esterne.

Nel contesto del framework unificato, l'incertezza si manifesta a due livelli distinti che è importante tenere separati. Il primo è l'**incertezza epistemica** — quella dovuta alla mancanza di dati o di esperienza — che si manifesta come varianza elevata della distribuzione posteriore $q_\phi(s_i|x_i)$ per un'osservazione x_i che il sistema non ha incontrato nelle stesse circostanze durante l'addestramento. Il secondo è l'**incertezza aleatoria** — quella intrinseca alla struttura stocastica del mondo — che si manifesta come varianza elevata del prior condizionale appreso $p_\theta(s|s_i,z)$ per una coppia (s_i, z) che corrisponde a una situazione in cui il target s è genuinamente non determinato dal contesto.

La distinzione fra questi due tipi di incertezza è fondamentale per la progettazione di sistemi AGI robusti. Un sistema che confonde incertezza epistemica e aleatoria potrebbe astenersi dall'agire in situazioni che richiedono azione (perché percepisce come «incerto» qualcosa che è semplicemente aleatorio ma gestibile) o potrebbe agire con eccessiva sicurezza in situazioni in cui la propria esperienza è insufficiente (perché la varianza del prior condizionale è bassa anche per osservazioni mai viste, purché simili a quelle viste). Nel Var-JEPA, questa distinzione emerge strutturalmente dalla differenza fra la varianza di $q_\phi(s_i|x_i)$ — che misura l'incertezza epistemica del codificatore — e la varianza di $p_\theta(s|s_i,z)$ − la varianza di $q_\phi(s|s_i,z,y)$ — che misura l'incertezza aleatoria residua dopo la predizione.

Formalmente, se si definisce l'incertezza totale sulla rappresentazione target come:

$\mathcal{U}$_totale(s | s_i, z) = Var_{q_ϕ}[s | s_i, z, y] + $\mathbb{E}$_{q_ϕ(s | s_i,z,y)}[(s − μ^s _θ(s_i,z))²]

il primo termine è la varianza della distribuzione posteriore del target — incertezza epistemica nella stima del target dato il contesto osservato — e il secondo termine è il bias al quadrato fra la stima media del target e la previsione del prior condizionale — incertezza aleatoria come distanza media fra ciò che il sistema osserva e ciò che il suo modello del mondo prevede. Questa decomposizione è l'analogo variazionale della legge della varianza totale e ha un'interpretazione cognitiva diretta: il primo termine misura quanto il sistema è sicuro di ciò che ha visto; il secondo misura quanto il suo modello del mondo è adeguato a spiegare ciò che ha visto.

18.9 I limiti della sintesi e le domande aperte

Qualsiasi tentativo onesto di costruire un quadro unificato deve concludersi con un riconoscimento esplicito delle sue lacune. Il framework proposto in questo capitolo è coerente sul piano formale e plausibile sul piano cognitivo, ma presenta limitazioni che nessuno dei tre paper di riferimento risolve, e che non possono essere superate attraverso la sola elaborazione teorica.

La più importante riguarda la **validazione empirica a scala**. LLM-JEPA è validato su modelli da 1B a 8B parametri; Var-T-JEPA è validato su dataset tabulari di dimensione moderata (al massimo 48.000 campioni). Il framework unificato richiederebbe addestramento su dataset multimodali di scala paragonabile a quella dei sistemi all'avanguardia come GPT-4 o PaLM-E, con miliardi di coppie (testo, immagine), (testo, codice), (testo, segnale propriocettivo), e così via. A quella scala, le interazioni fra i diversi termini della perdita unificata potrebbero manifestare comportamenti che i modelli a scala ridotta non anticipano.

La seconda limitazione riguarda la **natura del fondamento simbolico** nell'architettura proposta. Il prior condizionale appreso p_θ(s | s_i,z) implementa una forma di grounding che è **implicita** e **distribuita**: il significato dei simboli emerge dalla struttura delle relazioni predittive nello spazio latente, non da un'assegnazione esplicita di significato attraverso regole o ontologie. Mumuni e Mumuni discutono anche approcci al grounding esplicito — knowledge graphs, ontologie, active exploration attraverso reinforcement learning — che l'architettura Var-JEPA multimodale non include direttamente. L'integrazione di strutture simboliche esplicite con l'inferenza variazionale distribuita rimane un problema aperto che nessuno dei tre paper affronta sistematicamente.

La terza limitazione riguarda il **ragionamento controfattuale**. Sia Var-JEPA che LLM-JEPA operano al livello 1 e parzialmente al livello 2 della gerarchia causale di Pearl — associazione e intervento — ma nessuno dei due si avventura nel livello 3, quello controfattuale, che richiede di ragionare su «cosa sarebbe successo se le condizioni fossero state diverse». Il DAG del processo generativo Var-JEPA è compatibile con il ragionamento controfattuale — la teoria degli SCM di Pearl è formulata precisamente in termini di grafi causali della stessa struttura — ma l'implementazione concreta di operazioni di intervento (do-calculus) e di controfattuale nel framework variazionale richiederebbe modifiche architetturali significative che esulano dallo scope dei paper qui analizzati.

Queste lacune non infirmano la coerenza del framework proposto. Esse indicano semmai il programma di ricerca che una vera sintesi richiederebbe: validazione empirica a scala, integrazione di strutture simboliche esplicite, e estensione del formalismo variazionale al ragionamento causale di livello 3. Sono questioni aperte, non obiezioni insuperabili — e la loro identificazione precisa è già, di per sé, un contributo alla chiarezza del dibattito.

17.10 Prospettiva informazionale: JEPA, AMI e la gerarchia causale di Pearl

Una prospettiva che unifica AMI, Var-JEPA e LLM-JEPA è quella informazionale basata sulla gerarchia di comprensione di Pearl (associazione, intervento, controfattuale). I diversi sistemi si collocano a livelli diversi della gerarchia non perché abbiano architetture fondamentalmente diverse, ma perché ottimizzano diversi obiettivi informativi. LLM-JEPA opera prevalentemente al livello 1 (associazione): il termine cosine similarity tra rappresentazioni di viste co-occorrenti apprende correlazioni statistiche senza esplicitare relazioni causali. Var-JEPA con il suo DAG causale esplicito si avvicina al livello 2 (intervento): la struttura (s_x → s_y, z → s_y) codifica un'ipotesi causale sul modo in cui il contesto influenza il target. Il framework AMI nella sua visione completa mira al livello 3 (controfattuale): il world model deve permettere di ragionare su "cosa sarebbe successo se avessi agito diversamente?".

Il gap tra i livelli 1-2 (dove operano i sistemi JEPA attuali) e il livello 3 (richiesto per l'AGI genuina) non è colmabile con aumenti di scala dei dati di addestramento senza modifiche architetturali che incorporino strutture causali esplicite. La distinzione tra distribuzione osservazionale e interventistica — $P(s_y \mid s_x)$ vs $P(s_y \mid do(s_x = \tilde{s}_x))$ — rimane il confine che separa i sistemi JEPA attuali dall'intelligenza causale genuina, confine che richiede la presenza di variabili esogene strutturali e meccanismi di intervento che nessuno dei tre framework analizzati incorpora.

17.11 Confronto information-teorico: information bottleneck e capacità di rappresentazione

Il principio Information Bottleneck (IB, Tishby et al. 2000) fornisce una metrica comune per confrontare le architetture. La rappresentazione ottimale z di x per predire y deve massimizzare $I(z; y)$ minimizzando $I(z; x)$. Il piano informativo $(I(z; x), I(z; y))$ descrive il trade-off tra compressione e predittività. Per Var-JEPA, la decomposizione ELBO surgery mostra che $I_{q_\varphi}(x; s_x) = E_x[KL(q_\varphi(s_x|x) \,||\, q_\varphi(s_x))]$ è il termine di informazione mutua regolarizzato dal termine KL dell'ELBO. Il coefficiente $\alpha^{KL}_{s_x}$ controlla la posizione nel piano informativo: un valore alto sposta il sistema verso sinistra (più compressione), un valore basso verso destra. Per LLM-JEPA, l'obiettivo JEPA sposta il sistema verso il basso-destra nel piano informativo — più compressione della forma superficiale, più informazione sulla struttura semantica — spiegando sia la migliore generalizzazione sia la resistenza all'overfitting.

Capitolo 19 — Limiti residui: ciò che nessuno dei tre contributi risolve

19.1 Il silenzio epistemico condiviso

Ogni sviluppo scientifico di rilievo porta con sé, incorporata nella propria struttura argomentativa, la mappa implicita di ciò che ancora non sa affrontare. I tre contributi esaminati nei capitoli precedenti — la rassegna di Mumuni e Mumuni sull'intelligenza artificiale generale con modelli di linguaggio di grandi dimensioni, la proposta di Huang, LeCun e Balestriero che trapiantia l'architettura JEPA nel dominio linguistico, e la reinterpretazione variazionale di Gögl e Yau — costituiscono, presi separatamente, altrettanti passi di un cammino; considerati nella loro interazione, tuttavia, rivelano una topografia di lacune condivise che nessuno di essi, né singolarmente né in combinazione, riesce a colmare. Il presente capitolo non mira a sminuire il valore intrinseco di questi lavori, bensì a rendere esplicita tale topografia, comprendendo sia le omissioni deliberatamente dichiarate dagli autori stessi sia quelle che emergono soltanto dal confronto trasversale.

Il punto di partenza più onesto è riconoscere che i tre paper operano su piani di astrazione distinti e che la loro messa a confronto non è priva di rischi ermeneutici. Mumuni e Mumuni si collocano al livello dei principi cognitivi fondativi — incarnazione, radicamento simbolico, causalità, memoria — e si interrogano su quale architettura concettuale debba sottendere un sistema capace di intelligenza generale. Huang, LeCun e Balestriero operano invece a un livello tecnico molto più ravvicinato, concentrandosi su come modificare la funzione obiettivo del preaddestramento e del fine-tuning per i modelli di linguaggio. Gögl e Yau occupano una posizione intermedia: riformulano il paradigma JEPA all'interno di un apparato variazionale rigoroso, e lo istanziano su dati tabulari, ma la loro ambizione dichiarata rimane essenzialmente di carattere teorico-fondazionale più che empirica su larga scala. Eppure, malgrado questa eterogeneità, i tre lavori condividono un insieme di problemi irrisolti che vale la pena enumerare con precisione analitica.

19.2 Il radicamento simbolico rimane un postulato, non una conseguenza

Il problema del radicamento simbolico — quello che Harnad formulò negli anni Novanta come la questione di come i simboli interni a un sistema computazionale possano acquisire un significato che non sia puramente relazionale ma ancorato al mondo fisico — attraversa i tre contributi in modo asimmetrico. Mumuni e Mumuni vi dedicano una trattazione sistematica nel quarto capitolo della loro rassegna, illustrando sei classi principali di strategie: i grafi di conoscenza, il prompting guidato da ontologie, l'incorporamento nello spazio vettoriale, l'esplorazione attiva e l'interazione con l'ambiente, il radicamento mediante modelli generativi, e infine il ricorso a basi di conoscenza esterne attraverso la generazione aumentata dal recupero. Tuttavia, gli stessi autori ammettono esplicitamente che nessuna di queste tecniche realizza un radicamento nel senso pieno del termine: le rappresentazioni vettoriali sono scalabili ma opache e inclini a instaurare connessioni spurie; i grafi di conoscenza sono

interpretabili ma difficili da costruire e da mantenere aggiornati; l'esplorazione attiva produce connessioni fisicamente plausibili ma non può operare su concetti puramente astratti.

LLM-JEPA e Var-JEPA, per parte loro, non affrontano questo problema nemmeno tangenzialmente. L'obiettivo JEPA — nella sua declinazione linguistica come in quella variazionale — consiste nel garantire che la rappresentazione di un contesto possa predire la rappresentazione di un obiettivo correlato, minimizzando una distanza nello spazio degli embedding. Formalmente, nel caso di LLM-JEPA la perdita composita assume la forma:

$$\mathcal{L}_{\text{LLM-JEPA}} = \Sigma_{\ell=2}^{L} \mathcal{L}_{\text{LLM}}(\text{Text}_{1:\ell-1}, \text{Text}_{\ell}) + \lambda \cdot d(\text{Pred}(\text{Enc}(\text{Text})), \text{Enc}(\text{Code}))$$

dove d è tipicamente la similarità coseno negata, Pred(·) è il predittore con pesi legati all'LLM stesso, e $\lambda \geq 0$ è un iperparametro di bilanciamento. Nel caso di Var-JEPA la struttura si arricchisce di cinque termini variazionali:

$$\mathcal{L}_{\text{ELBO}}(x, y; \theta, \varphi) = \alpha^{\text{rec}} \cdot (-\mathbb{E}_{q_\varphi(s_x|x)}[\log p_\theta(x|s_x)]) + \alpha^{\text{gen}} \cdot (-\mathbb{E}_{q_\varphi(s_x,z,s_y|x,y)}[\log p_\theta(y|s_y)]) + \alpha^{\text{KL}}_{s_x} \cdot \text{KL}(q_\varphi(s_x|x) \| p(s_x)) + \alpha^{\text{KL}}_{z} \cdot KL(q_\varphi(z|s_x) \| p(z)) + a^{KL}\{s_y\} \cdot \text{KL}(q_\varphi(s_y|s_x, z, y) \| p_\theta(s_y|s_x, z))$$

In entrambe le formulazioni le rappresentazioni s_x, s_y e z vivono in uno spazio vettoriale continuo e vengono apprese esclusivamente sulla base della struttura statistica delle coppie di dati d'ingresso. Non vi è alcun meccanismo che costringa tali rappresentazioni a corrispondere a entità o relazioni nel mondo fisico: la similarità coseno fra Enc(Text) e Enc(Code) misura una prossimità geometrica nello spazio dei parametri, non una correlazione con il referente esterno. Il predictor di LLM-JEPA, implementato aggiungendo un token speciale [PRED] in coda all'input e sfruttando il meccanismo di auto-attenzione dell'LLM con pesi condivisi, apprende a mappare la rappresentazione del testo verso quella del codice sorgente corrispondente, ma questo apprendimento non richiede — né garantisce — che le coordinate dello spazio latente abbiano un'interpretazione semantica stabile rispetto al dominio di applicazione.

Gögl e Yau sono consapevoli di questa limitazione strutturale: quando osservano che il JEPA classico «è suscettibile di collasso rappresentazionale», si riferiscono al rischio che gli encoder apprendano a produrre vettori costanti o concentrati su sottospazi di dimensione trascurabile, minimizzando la perdita senza acquisire alcuna conoscenza utile. La Var-JEPA risolve questo problema attraverso la regolarizzazione variazionale — i termini KL rispetto ai prior standard-normali per s_x e z impediscono che le distribuzioni posteriori collassino — ma questa soluzione opera esclusivamente a livello della geometria dello spazio latente, non a livello del contenuto semantico delle rappresentazioni. La condizione:

$$\mathbb{E}_x[\text{KL}(q_\varphi(s_x|x) \| \mathcal{N}(0, I))] = \text{KL}(q_\varphi(s_x) \| \mathcal{N}(0, I)) + I_{q_\varphi}(x; s_x)$$

— che gli autori derivano applicando la cosiddetta «ELBO surgery» di Hoffman e Johnson — garantisce che la distribuzione aggregata degli embedding sia prossima a una gaussiana isotropa, il che è favorevole per i probe di classificazione lineare a valle, ma non ha alcuna implicazione diretta sul radicamento dei simboli nel mondo esterno.

19.3 La causalità assente dallo spazio di predizione

La questione causale è forse la più profonda fra quelle lasciate aperte dai tre contributi, perché tocca simultaneamente la struttura della perdita d'addestramento, la natura delle rappresentazioni apprese e la capacità del modello di generalizzare a distribuzioni shifted rispetto a quella di addestramento. Mumuni e Mumuni dedicano l'intera quinta sezione della loro rassegna al tema della causalità nei modelli di linguaggio di grandi dimensioni, riprendendo la tassonomia di Pearl in tre livelli — associazione, intervento e controfattuale — e documentando come i sistemi allo stato dell'arte rimangano fondamentalmente fermi al primo livello. I modelli addestrati su grandi corpora testuali apprendono distribuzioni condizionali $P(Y|X)$ che catturano correlazioni statistiche, ma non dispongono di un modello strutturale del tipo:

$$Y := f(X, U_Y), \quad X := g(U_X)$$

dove U_X e U_Y sono variabili esogene (rumori latenti indipendenti), f e g sono meccanismi causali distinti, e l'operatore «:=» denota l'assegnazione strutturale piuttosto che la semplice uguaglianza. In assenza di tale struttura, le previsioni del modello rimangono fragili rispetto a qualunque intervento $do(X = x)$ che rompa il legame osservazionale fra X e Y.

LLM-JEPA non menziona affatto il problema causale. La perdita composita che propone è puramente associativa: apprende che certe codifiche di testo devono essere vicine nello spazio degli embedding a certe codifiche di codice sorgente, sulla base di coppie osservazionali (Text, Code). Non vi è alcuna garanzia che questa prossimità rifletta una relazione causale — potrebbe semplicemente catturare la covariazione statistica indotta dal fatto che certi problemi matematici in linguaggio naturale tendono a comparire assieme a certe soluzioni in Python all'interno dei dati di addestramento. La capacità di LLM-JEPA di migliorare le prestazioni su benchmark come GSM8K — rilevata dagli autori — non può essere interpretata come evidenza di comprensione causale; il miglioramento è coerente con una più efficace compressione statistica del pattern di co-occorrenza (problema, soluzione).

La situazione di Var-JEPA è analoga, sebbene la natura probabilistica del suo framework apra almeno concettualmente uno spazio per riflessioni causali. Il prior condizionato $p_\theta(s_y | s_x, z)$ — che gli autori interpretano come il «predictor causale» della Var-JEPA — è una gaussiana i cui parametri $\mu^{s_y}_\theta(s_x, z)$ e $\Sigma^{s_y}_\theta(s_x, z)$ sono calcolati da reti neurali addestrate a minimizzare la divergenza rispetto alla distribuzione posteriore $q_\varphi(s_y | s_x, z, y)$. La minimizzazione di questa divergenza KL è equivalente alla massimizzazione della verosimiglianza variazionale condizionata, il che significa che $p_\theta(s_y | s_x, z)$ apprende la distribuzione osservazionale di s_y dato s_x e z, non la distribuzione interventistica $P(s_y | do(s_x = \tilde{s}_x))$. Per comprendere la differenza, si consideri che la distribuzione osservazionale è:

$$p(s_y | s_x) = \int p(s_y | s_x, u) \cdot p(u | s_x)\, du$$

mentre quella interventistica, ottenuta mediante il calcolo do-operator di Pearl, è:

$$P(s_y | do(s_x = \tilde{s}_x)) = \int p(s_y | \tilde{s}_x, u) \cdot p(u)\, du$$

Le due quantità coincidono soltanto se s_x e u sono indipendenti, ovvero in assenza di confonditori latenti. Il modello generativo di Var-JEPA non impone questa struttura: la variabile ausiliaria z, condizionata su s_x attraverso q_φ(z|s_x), può assorbire informazioni sia sulle cause di s_x sia su quelle di s_y, rendendo di fatto impossibile la separazione intervento/osservazione all'interno del framework come proposto.

19.4 Il problema della costruzione delle viste: una limitazione strutturale di LLM-JEPA

Una delle assunzioni più delicate di LLM-JEPA — e che Huang, LeCun e Balestriero riconoscono esplicitamente — è la necessità di disporre, per ciascun campione di addestramento, di almeno due «viste» non banali dello stesso contenuto sottostante. Gli autori definiscono la coppia (Text, Code) come un caso paradigmatico: il testo in linguaggio naturale che descrive un problema e il codice sorgente che ne costituisce la soluzione sono «due viste della stessa conoscenza sottostante». Questa intuizione è teoricamente solida — ed è esattamente ciò che giustifica l'applicazione dell'obiettivo JEPA nel dominio del linguaggio — ma la sua implementazione pratica è condizionata alla disponibilità di dataset strutturati in coppie di questo tipo.

Gli autori sperimentano su NL-RX-SYNTH e NL-RX-TURK (espressioni regolari da descrizioni in linguaggio naturale), Spider (SQL da query in linguaggio naturale), GSM8K (soluzioni numeriche da problemi matematici in testo) e HellaSwag (completamento di contesto). Questi benchmark condividono la proprietà di fornire, per definizione del task, coppie in cui le due modalità sono distinte ma semanticamente co-referenziali. Tuttavia, come gli autori stessi notano, «sviluppare un meccanismo analogo all'augmentazione dei dati nella visione consentirebbe di applicare gli obiettivi JEPA a qualunque dataset»: questa affermazione è al tempo stesso un'ammissione di un limite attuale e un programma di ricerca futuro, non una soluzione già disponibile.

Il punto critico è il seguente. Nella visione, le viste multiple si costruiscono applicando trasformazioni invarianti all'immagine originale — ritagli, rotazioni, cambiamenti di luminosità — in modo che due viste dello stesso oggetto mantengano le proprietà semanticamente rilevanti pur differendo superficialmente. La scelta di quale trasformazione sia «invariante» nel senso giusto è guidata da conoscenza di dominio accumulata decennalmente nella comunità. Per il linguaggio naturale non esiste un analogo maturo: la parafrasi è una candidata ovvia, ma due parafrasi possono differire in sfumature pragmatiche che sono semanticamente rilevanti (ad esempio, la distinzione fra enunciati assertivi e interrogativi). Gli autori esplorano brevemente l'uso del dataset cestwc/paraphrase, costruendo la perdita JEPA facendo predire la i-esima parafrasi dalla (i+1)-esima, e riscontrano miglioramenti statisticamente significativi su Rotten Tomatoes e Yelp dopo fine-tuning; ma questa sperimentazione è riconosciuta come preliminare, e il problema della costruzione automatica di viste non banali per testi arbitrari rimane sostanzialmente aperto.

Var-JEPA non affronta direttamente questo problema — il suo contributo è di riformulare l'obiettivo JEPA in termini variazionali piuttosto che di estenderne il dominio di applicabilità — ma ne eredita implicitamente la dipendenza: anche nella Var-JEPA applicata a dati tabulari (Var-T-JEPA), la coppia (x, y) è costruita mediante mascheramento casuale di feature a livello

di colonne, e la validità dell'obiettivo dipende dal fatto che le feature mascherate nel contesto e quelle mascherate nel target condividano informazione sufficiente da rendere la predizione non banale ma nemmeno impossibile. Il range dei parametri di mascheramento — M_{ctx} ∈ [r^{min}{*ctx*} · *D*{feat}, r^{max}{*ctx*} · *D*{feat}] e analogamente per M_{trg} — è scelto manualmente per ciascun dataset, e gli autori riferiscono che la scelta di questi iperparametri influenza sensibilmente le prestazioni a valle.

19.5 La scalabilità di Var-JEPA: da tabular a immagini e video, un salto non dimostrato

Gögl e Yau concludono il loro articolo con l'osservazione che «il lavoro futuro include la scalabilità alla visione e al video, e l'estensione a contesti in cui le osservazioni target sono assenti al momento del test». Questa onestà programmatica delimita con precisione l'orizzonte attuale del contributo: Var-T-JEPA è una prova di concetto implementata in un dominio — quello dei dati tabulari — che, per quanto impegnativo nella pratica, richiede risorse computazionali notevolmente inferiori rispetto alla visione su larga scala.

La questione non è meramente tecnica. Il framework Var-JEPA richiede, per ciascun campione x, di calcolare quattro passaggi in avanti distinti: il context encoder q_φ(s_x|x), l'auxiliary encoder q_φ(z|s_x), il target posterior q_φ(s_y|s_x, z, y), e il predictive model p_θ(s_y|s_x, z). Ciascuno di questi passaggi coinvolge reti transformer separate (o varianti di esse), con la conseguenza che il costo computazionale totale è un multiplo significativo di quello richiesto da un semplice encoder. In I-JEPA, la versione originale per immagini, il predictor è deliberatamente leggero — pochi strati di transformer con dimensione nascosta ridotta — e il target encoder è aggiornato tramite media mobile esponenziale (EMA) dei parametri del context encoder, senza propagazione del gradiente. Nella Var-JEPA, invece, il target posterior q_φ(s_y|s_x, z, y) deve elaborare l'intera osservazione target y per parametrizzare la distribuzione variazionale: questo implica che l'encoder del target riceva sia i token delle feature mascherate del target sia i token latenti del contesto e dell'ausiliario come input aggiuntivi — un overhead che, nel caso di immagini ad alta risoluzione o video, potrebbe risultare proibitivo senza interventi architetturali specifici.

Gli autori abbozzano in appendice una possibile estensione alla visione, suggerendo di (i) tokenizzare le immagini in sequenze di patch, (ii) definire viste context e target mascherate analogamente a I-JEPA, (iii) parametrizzare q_φ(s_x|x) e q_φ(s_y|s_x, z, y) con encoder basati su Vision Transformer, e (iv) aggiungere un decoder p_θ(y|s_y) — ad esempio una gaussiana sul valore dei pixel o una distribuzione su token discreti — per ottenere l'obiettivo ELBO. La fattibilità di questo schema non è però empiricamente verificata nell'articolo, e rimane quindi un'ipotesi di lavoro non consolidata.

19.6 Il deficit di incarnazione: un'assenza strutturale

La rassegna di Mumuni e Mumuni si apre con una considerazione fondamentale che nessuno dei contributi tecnici successivi riesce a contestare: l'incarnazione — intesa come la capacità del sistema intelligente di percepire il mondo attraverso un corpo fisico, di agire su di esso in modo intenzionale e di ricevere retroazione sensorimotoria — non è un complemento

opzionale dell'intelligenza artificiale generale, ma una precondizione per il tipo di apprendimento situato che genera conoscenza robusta e trasferibile. La triplice unità mente-corpo-ambiente, che gli studi di neuroscienza cognitiva postulano come costitutiva dei processi cognitivi biologici, non trova alcun corrispettivo nell'architettura di LLM-JEPA né in quella di Var-JEPA.

Questo non è un difetto contingente, rimediabile con un aggiustamento marginale: è una conseguenza della struttura stessa degli obiettivi di addestramento. LLM-JEPA apprende relazioni nello spazio degli embedding di testi e codice; Var-JEPA apprende a predire distribuzioni latenti di feature tabulari da sottoinsieme di feature mascherate. In entrambi i casi il sistema non sperimenta le conseguenze fisiche delle proprie azioni nell'ambiente, non aggiorna le proprie rappresentazioni sulla base di interazioni deliberate con oggetti e fenomeni, non possiede un modello delle proprie capacità e limitazioni sensorimotorie. Mumuni e Mumuni distinguono quattro aspetti dell'intelligenza incarnata — la consapevolezza degli obiettivi (goal-awareness), la consapevolezza di sé (self-awareness), la consapevolezza situazionale (situational-awareness) e l'azione deliberata — e documentano come ciascuno di essi richieda forme di apprendimento e di interazione che i modelli linguistici allo stato dell'arte non riescono a sostenere in modo robusto e generalizzabile.

La correlazione con JEPA è sottile ma rilevante. L'aspirazione originaria di LeCun, nel documento programmatico del 2022 da cui il framework JEPA trae ispirazione, era di costruire un modello del mondo (world model) capace di pianificare e agire in ambienti continui e incerti. In questa visione, il JEPA è un componente di un'architettura cognitiva più ampia che include un modulo di percezione, un modulo di azione, un modulo di memorizzazione e un modulo di ottimizzazione dell'azione. LLM-JEPA e Var-JEPA affrontano esclusivamente la componente percettivo-rappresentazionale di questa architettura — il problema di apprendere rappresentazioni latenti ricche e strutturate — senza toccare i moduli di azione e di pianificazione. Si tratta di un contributo genuino, ma circoscritto a una porzione del problema complessivo.

19.7 Memoria, apprendimento continuo e la persistenza del dimenticatoio catastrofico

La sesta sezione della rassegna di Mumuni e Mumuni affronta il tema della memoria nei modelli di linguaggio con una sistematicità che contrasta nettamente con la totale assenza del tema nei due contributi tecnici. La memoria è lì descritta come il meccanismo attraverso cui la conoscenza acquisita mediante incarnazione, radicamento e ragionamento causale viene consolidata, organizzata e resa disponibile per impieghi futuri. Gli autori distinguono tre livelli — memoria sensoriale, memoria di lavoro e memoria a lungo termine — e suddividono quest'ultima in semantica, episodica e procedurale. Per ciascuna forma identificano le corrispondenti implementazioni nei sistemi artificiali: le strutture di conoscenza in forma di grafi o ontologie per la memoria semantica, la finestra contestuale del transformer come analogo della memoria di lavoro, e le tecniche di fine-tuning specializzato o di recupero aumentato come approssimazioni della memoria episodica e procedurale.

LLM-JEPA non propone alcuna modifica all'architettura della memoria del modello linguistico sottostante. La perdita composita $\mathcal{L}_{\text{LLM-JEPA}}$ viene calcolata e minimizzata su batch di dati in modo stazionario: ogni passaggio di addestramento aggiorna i parametri dell'encoder e del predictor sulla base del gradiente rispetto al batch corrente, senza alcun meccanismo di protezione dei parametri già specializzati su compiti precedenti. Il problema del dimenticatoio catastrofico — che Goodfellow e colleghi avevano caratterizzato sperimentalmente già nel 2013 come la tendenza delle reti neurali a perdere le conoscenze acquisite su compiti precedenti quando vengono ri-addestrate su compiti nuovi — rimane quindi pienamente operativo nell'architettura di LLM-JEPA. Gli autori osservano che LLM-JEPA «resiste all'overfitting» rispetto al fine-tuning standard — un miglioramento evidenziato nelle curve di accuratezza in funzione delle epoche di addestramento su NL-RX-SYNTH — ma questa resistenza concerne la generalizzazione al test set dello stesso task, non la ritenzione di competenze acquisite su task precedenti.

Var-JEPA si trova nella medesima situazione. L'obiettivo ELBO viene ottimizzato su un dataset fisso, e gli autori non propongono alcun meccanismo per aggiornare il modello in modo incrementale mantenendo le rappresentazioni già apprese. Anzi, la presenza dei termini KL di regolarizzazione — che spingono le distribuzioni posteriori verso prior gaussiani standard — introduce una pressione verso la compressione dell'informazione che potrebbe aggravare la perdita di conoscenza pregressa in scenari di apprendimento sequenziale: se un nuovo dataset impone una distribuzione di s_x radicalmente diversa da quella appresa su dataset precedenti, la regolarizzazione verso $\mathcal{N}(0, I)$ forza il modello ad adottare la nuova distribuzione, cancellando le strutture latenti precedentemente acquisite.

Un aspetto particolarmente degno di nota è che la capacità di quantificazione dell'incertezza offerta da Var-JEPA — attraverso la varianza della distribuzione posteriore $q_\varphi(s_y | s_x, z, y)$ — potrebbe in linea di principio segnalare l'arrivo di input out-of-distribution, fornendo così un indicatore utile per la gestione della memoria in contesti di apprendimento continuo. Tuttavia, gli autori non sviluppano questa possibilità: la valutazione selettiva che propongono — scartare il 10%, 20% o 50% dei campioni di test più incerti prima di calcolare l'accuratezza — è un'applicazione passiva dell'incertezza latente a un problema di selezione del sottoinsieme di test, non un meccanismo attivo di gestione dell'apprendimento. La distanza fra questa applicazione e un sistema di apprendimento continuo pienamente funzionale è considerevole.

19.8 La sensibilità agli iperparametri come limite operativo di entrambi i framework

Entrambi i contributi tecnici segnalano, con gradi diversi di enfasi, un problema di sensibilità agli iperparametri che va al di là di quanto tipicamente osservato nell'addestramento di modelli standard. Nel caso di LLM-JEPA, i due iperparametri aggiuntivi rispetto al baseline sono k — il numero di token [PRED] appesi all'input per costruire il predictor — e λ, il peso relativo del termine JEPA rispetto alla perdita LLM standard. La griglia di ricerca suggerita dagli autori, $(k, \lambda) \in \{0, 1, 2, 3, 4\} \times \{0.5, 1, 2, 4\}$, implica venti configurazioni distinte, ciascuna eseguita cinque volte con semi casuali diversi per permettere il t-test statistico: si tratta di cento run di addestramento per singola combinazione (modello, dataset) prima di poter scegliere la configurazione ottimale. Gli autori osservano esplicitamente: «non abbiamo

trovato alcuno schema su dove possa apparire la migliore accuratezza. Potrebbe trovarsi sia all'estremo alto sia all'estremo basso di λ o k. Inoltre, possono esserci cali e picchi in posizioni casuali».

Questa osservazione è di importanza non trascurabile: suggerisce che il paesaggio della perdita in funzione degli iperparametri è irregolare e non convesso, rendendo le strategie di ottimizzazione basate su gradienti o su ricerca bayesiana potenzialmente inefficaci. In pratica, il costo computazionale dell'ottimizzazione degli iperparametri per LLM-JEPA è proporzionale al quadruplo di quello già sostenuto per addestrare il baseline, poiché ciascun run di LLM-JEPA richiede due passaggi in avanti (uno per il testo, uno per il codice, con la maschera di attenzione personalizzata) invece di uno solo.

Var-JEPA introduce una complessità iperparametrica ancora maggiore. I pesi degli α per i cinque termini della perdita — α^{rec}, α^{gen}, α^{KL}{*s_x*}, *a*^{*KL*}*z*, *a*^{*KL*}{*s_y*} — *interagiscono in modo non lineare con le schedule di annealing dei termini KL (con tempi T^{anneal}*{s_x}, T^{anneal}*z*, *T*^{*anneal*}{s_y} configurabili separatamente), i parametri di mascheramento, la dimensionalità nascosta d, il numero di strati L, la dimensionalità del predictor, e così via. La Tabella 7 dell'articolo di Gögl e Yau elenca per ciascun dataset una configurazione specifica di questi iperparametri, frutto evidentemente di una tuning manuale o semi-automatica che non è documentata nel dettaglio. I test di sensibilità presentati in appendice mostrano che le prestazioni sono «generalmente robuste» rispetto a variazioni moderate dei pesi α e dei parametri architetturali, ma «alcune variazioni (ad esempio encoder significativamente più profondi o batch size più piccola) possono ridurre le prestazioni»: una valutazione che lascia aperto un margine di incertezza non trascurabile sul comportamento in configurazioni non testate.

19.9 L'integrazione fra incertezza variazionale e capacità generativa: un nodo irrisolto

Un limite più sottile, che emerge dall'accostamento fra LLM-JEPA e Var-JEPA, riguarda la possibilità di integrare la quantificazione dell'incertezza offerta dal framework variazionale con le capacità generative del modello linguistico. LLM-JEPA mantiene e potenzia la capacità generativa dell'LLM: la perdita composita preserva il termine $\mathcal{L}$_{LLM} di ricostruzione nel token space, garantendo che il modello rimanga capace di completare testi e generare risposte. Var-JEPA, per contro, è orientata alla produzione di embedding deterministici (medie delle distribuzioni posteriori) per compiti di classificazione a valle, e la sua capacità generativa — intesa come la produzione di osservazioni y a partire da contesti x attraverso la catena s_x → z → s_y → y — è presentata come una possibilità teorica (sezione 3.6 dell'articolo) ma non sfruttata nelle valutazioni sperimentali.

La domanda naturale che sorge dalla lettura comparata dei due articoli è se sia possibile costruire un sistema che sia simultaneamente (i) generativo nel senso di LLM-JEPA — capace di completare testi, tradurre problemi in codice, rispondere a domande — e (ii) variazionale nel senso di Var-JEPA — capace di quantificare l'incertezza sulla propria rappresentazione latente e di generalizzare in modo robusto a input out-of-distribution. La risposta teoricamente corretta è affermativa in linea di principio: la struttura della Var-JEPA è compatibile con un decoder linguistico al posto del decoder gaussiano per la ricostruzione di

y. Tuttavia, la realizzazione pratica di questo schema imporrebbe che il termine di generazione $\mathcal{L}_{gen}$ nella perdita ELBO sia calcolato su sequenze di token attraverso la cross-entropy autoregressiva — esattamente come $\mathcal{L}_{LLM}$ in LLM-JEPA — aggiungendo però i quattro passaggi variazionali aggiuntivi (context encoder stocastico, auxiliary encoder, target posterior e predictive prior) all'overhead già significativo di LLM-JEPA. Nessuno dei due articoli esplora questa direzione, e i due framework rimangono quindi su binari paralleli che non si incontrano.

19.10 La plausibilità fisica delle rappresentazioni: un criterio assente

La rassegna di Mumuni e Mumuni dedica ampio spazio — in particolare nel paragrafo 5.2.3 — ai modelli del mondo basati su fisica intuitiva (intuitive physics), osservando che la capacità umana di ragionare su fenomeni causali si basa su un modello implicito e approssimato delle leggi fisiche che governa le interazioni fra oggetti materiali. Sistemi come PhysicsX, PyBullet e Isaac Gym offrono ambienti simulati fisicamente plausibili in cui agenti incarnati possono apprendere questi modelli attraverso interazione. Gli autori argomentano che l'addestramento in ambienti basati su simulazioni fisiche consente agli agenti di generalizzare in modo più robusto rispetto all'apprendimento da dataset statici, perché le rappresentazioni apprese sono vincolate a essere coerenti con le regole dell'interazione fisica.

Né LLM-JEPA né Var-JEPA impongono alcun vincolo di questo tipo sulle proprie rappresentazioni. Lo spazio latente di entrambi è uno spazio vettoriale euclideo non strutturato: la geometria è determinata interamente dalla struttura statistica dei dati di addestramento e dalla forma della perdita. Non vi è nulla nella formulazione matematica di questi framework che impedisca alle rappresentazioni di codificare pattern fisicamente implausibili o addirittura contraddittori, purché tali pattern siano statisticamente frequenti nei dati. Var-JEPA include decodificatori ricostruttivi che forzano le rappresentazioni latenti a conservare informazione sufficiente per ricostruire le osservazioni originali, il che introduce un vincolo implicito di fedeltà; ma questo vincolo opera a livello dei dati di addestramento, non a livello delle leggi fisiche che governano il dominio sottostante.

19.11 La valutazione dell'intelligenza generale: un criterio sfuggente

Un ultimo punto di tensione condiviso dai tre contributi riguarda il problema di misurare il progresso verso l'intelligenza artificiale generale. Mumuni e Mumuni lo riconoscono esplicitamente nella sezione discussiva: «valutare e accertare quando l'intelligenza generale venga effettivamente raggiunta è un problema difficile». La performance su benchmark specifici — che costituisce il metro di valutazione di LLM-JEPA su NL-RX-SYNTH, GSM8K, Spider ed HellaSwag — misura la capacità di risolvere task con distribuzione di test simile a quella di addestramento, ma non cattura la flessibilità, la robustezza e la generalità che contraddistinguono l'intelligenza biologica. Un modello può ottenere accuratezze molto elevate su GSM8K e nondimeno fallire su varianti superficialmente diverse degli stessi problemi, o su problemi che richiedono la combinazione di conoscenze provenienti da domini distinti.

La stessa Var-JEPA valuta la qualità delle rappresentazioni attraverso probe di classificazione lineare e attraverso metriche di incertezza su dataset come MNIST e SIM — strumenti che misurano la struttura locale dello spazio latente ma non la capacità del sistema di generalizzare a compiti mai visti in precedenza. La selezione selettiva — scartare i campioni più incerti e misurare l'accuratezza sul sottoinsieme rimanente — è una metrica di coerenza fra incertezza e errore, non una metrica di intelligenza generale.

Questa convergente difficoltà di valutazione non è un difetto specifico dei tre contributi: è una conseguenza della giovinezza del campo e della mancanza di consenso su cosa costituisca un test convincente di intelligenza artificiale generale. Tuttavia, la sua persistenza attraverso tre lavori di orientamento molto diverso suggerisce che il problema valutativo non possa essere trattato come separato da quello teorico: senza una teoria più precisa di cosa significhi «comprendere» nel senso cognitivamente rilevante del termine, non è possibile costruire benchmark che misurino in modo affidabile il grado di comprensione raggiunto da un sistema artificiale.

19.12 Riepilogo strutturato dei vuoti condivisi

Il percorso analitico condotto nei paragrafi precedenti consente di delineare con maggiore nitidezza la mappa dei limiti condivisi dai tre contributi. Il radicamento simbolico rimane un obiettivo dichiarato ma non realizzato: né LLM-JEPA né Var-JEPA forniscono un meccanismo che colleghi le coordinate dello spazio latente a referenti nel mondo esterno, e le strategie documentate da Mumuni e Mumuni per risolvere il problema sono tutte parziali e soggette a compromessi non risolti. La causalità è assente a livello fondazionale: le perdite di addestramento di entrambi i framework JEPA sono puramente associative, e la struttura probabilistica di Var-JEPA — pur aprendo uno spazio concettuale per ragionamenti causali — non impone vincoli di separabilità o di invarianza agli interventi che sarebbero necessari per un'inferenza causale genuina. La costruzione delle viste rimane un problema aperto per LLM-JEPA, limitandone l'applicabilità ai dataset con struttura naturale a due modalità complementari. La scalabilità di Var-JEPA oltre i dati tabulari è prospettata come lavoro futuro ma non dimostrata. L'incarnazione è strutturalmente assente da entrambi i framework tecnici. L'apprendimento continuo non è affrontato né dall'uno né dall'altro. L'integrazione fra incertezza variazionale e capacità generativa rimane un programma di ricerca non avviato. La plausibilità fisica delle rappresentazioni non è vincolata da nessuno dei due obiettivi di addestramento. La valutazione dell'intelligenza generale sfugge agli strumenti di misurazione disponibili.

Non si tratta di limiti contingenti, superabili con più dati o più parametri. Sono, nella loro maggior parte, conseguenze di scelte architetturali e di ipotesi epistemologiche che definiscono l'identità stessa dei framework proposti. Riconoscerli con precisione è il primo passo verso la formulazione di ipotesi di ricerca genuinamente nuove, alle quali il capitolo conclusivo sarà dedicato.

19.13 I teoremi di impossibilità come diagnosi strutturale dei limiti residui

[Integrazione da Rapp-JEPA, Cap. XI] Il Capitolo 19 ha identificato le lacune condivise dai tre contributi: il radicamento simbolico rimane un postulato, la causalità è assente a livello fondazionale, l'incarnazione è strutturalmente irrisolta. Queste diagnosi rischiano però di essere lette come deficit contingenti, correggibili con più dati o architetture più sofisticate. La monografia gemella Rapp-JEPA (Cap. XI) dimostra formalmente che non è così: si tratta di limiti strutturali del paradigma computazionale, dimostrabili attraverso tre argomenti di impossibilità che vale la pena richiamare sinteticamente per completare il quadro dei limiti residui.

Il primo è l'argomento modello-teorico di Putnam sull'impossibilità del radicamento del riferimento. Per qualsiasi sistema simbolico fisico, qualunque insieme di relazioni sintattiche ammette interpretazioni radicalmente diverse che lo soddisfano egualmente: la funzione M che mappa simboli a referenti nel mondo non è determinata dalla struttura interna del sistema. Questo vale a fortiori per i sistemi JEPA: le rappresentazioni latenti s_x, s_y, z sono determinate esclusivamente dalla struttura statistica delle coppie di dati, e nessun meccanismo interno al framework può ancorare le loro coordinate a proprietà specifiche del mondo fisico. Il secondo è l'argomento di Kripke sull'impossibilità di acquisizione della normatività per sola correzione statistica: il predittore di LLM-JEPA apprende a mappare Enc(Text) verso Enc(Code) in modo statisticamente consistente, ma questa consistenza non implica comprensione delle regole che governano la relazione testo-codice nel senso rilevante per la generalizzazione fuori distribuzione. Il terzo è la convergenza del teorema di indefinibilità di Tarski e del teorema di incompletezza di Gödel: qualunque sistema formale sufficientemente espressivo non può contenere una definizione interna di verità per i propri enunciati, e contiene enunciati veri che non può dimostrare. I framework JEPA operano all'interno di tale classe di sistemi: la distinzione fra inferenza valida e inferenza apparentemente valida non è decidibile dall'interno del sistema stesso.

Questi tre argomenti non invalidano i risultati empirici documentati nei capitoli precedenti — i miglioramenti in accuracy, calibrazione dell'incertezza e resistenza all'overfitting sono genuini — ma chiariscono il piano epistemico su cui operano: si tratta di progressi nella competenza funzionale, non di avvicinamenti misurabili alla comprensione strutturalmente radicata. La distinzione orienta le risorse di ricerca verso direzioni che possono in principio colmare i limiti — incarnazione attiva, apprendimento causale strutturale, integrazione multimodale con vincoli fisici — anziché verso quelle che li aggirano ridefinendo il problema.

Capitolo 20 — Verso un'architettura cognitiva ibrida: ipotesi di ricerca future

20.1 Il punto di partenza: ciò che i tre contributi suggeriscono congiuntamente

Ogni limite ben circoscritto contiene implicitamente la forma del passo successivo. Il Capitolo 19 ha tracciato con precisione analitica la topografia delle lacune condivise dai tre contributi; il compito del presente capitolo è trasformare quella mappa negativa in un programma di ricerca positivo, ricavando le ipotesi di lavoro future direttamente da ciò che gli autori stessi suggeriscono — nei paragrafi conclusivi, nelle appendici, nelle note a piè di pagina — e da ciò che la struttura matematica dei loro framework rende logicamente praticabile senza richiedere salti arbitrari verso l'ignoto.

Il punto di partenza più onesto consiste nel riconoscere che i tre contributi non si limitano a descrivere i propri risultati: ognuno abbozza, con gradi di dettaglio diversi, le direzioni che considera più promettenti per il lavoro successivo. Huang, LeCun e Balestriero dichiarano che «sviluppare un meccanismo analogo all'aumentazione dei dati nella visione consentirebbe di applicare gli obiettivi JEPA a qualunque dataset» e che «l'utilizzo del loss dropout combinato con la maschera di attenzione personalizzata offre prospettive positive per scalare ulteriormente LLM-JEPA al preaddestramento a piena scala con un overhead computazionale minimo». Gögl e Yau annunciano che «il lavoro futuro include la scalabilità alla visione e al video, e l'estensione a contesti in cui le osservazioni target sono assenti al momento del test, rendendo centrale la generazione condizionata». Mumuni e Mumuni concludono che «il progresso continuo verso l'AGI richiederà paradigmi fondamentalmente nuovi per progettare LLM che implementino tutti questi principi in modo unificato», e che tale unificazione «richiederà filosofie progettuali che integrino modelli di apprendimento profondo con tecniche neuro-simboliche». Questi tre enunciati — uno tecnico-operativo, uno teorico-architetturale, uno filosofico-programmatico — delineano insieme il perimetro di un'agenda di ricerca coerente che il presente capitolo si propone di articolare con la massima precisione consentita dai materiali disponibili.

20.2 La costruzione principiata delle viste per LLM-JEPA: verso un'aumentazione semantica del linguaggio

La dipendenza di LLM-JEPA dalla disponibilità di coppie di viste naturalmente distinte — (testo, codice), (domanda, risposta), (contesto, completamento) — è al tempo stesso la fonte della sua eleganza concettuale e il principale vincolo alla sua generalizzabilità. Il framework funziona perché tali coppie forniscono gratis ciò che nella visione viene costruito artificialmente mediante trasformazioni geometriche: due rappresentazioni dello stesso contenuto sottostante che differiscono nella superficie ma concordano nella sostanza. La domanda che l'esperimento con le parafrasi del dataset cestwc/paraphrase pone in modo indiretto è se sia possibile costruire questo tipo di viste in modo automatico e sistematico per testi privi di una struttura naturale a due modalità.

La risposta che Huang e colleghi abbozzano ma non sviluppano è affermativa, e merita un'esplorazione più approfondita. Nel dominio visivo, le trasformazioni considerate

invarianti — ritagli casuali, capovolgimenti orizzontali, variazioni di colore, sfocature gaussiane — sono state selezionate sulla base della conoscenza che il sistema visivo umano è robusto rispetto a tali perturbazioni. L'analogo linguistico richiederebbe di identificare le trasformazioni rispetto alle quali la comprensione semantica è invariante: la sinonimia lessicale, la parafrasi sintattica, la traduzione e ritraduzione in una lingua pivot, la rimozione di elementi ridondanti o la loro espansione parafrastica, la variazione del registro stilistico senza alterazione del contenuto proposizionale.

Ciascuna di queste trasformazioni può essere formalizzata come un operatore T_i che mappa un testo x verso una sua perturbazione x' tale che il contenuto proposizionale — la struttura semantica rilevante per il task — sia preservato, mentre la forma superficiale sia modificata. In questo schema, la perdita LLM-JEPA generalizzerebbe a:

$$\mathcal{L}_{\text{LLM-JEPA}}^{\text{aug}} = \Sigma_{\ell=2}^{L} \mathcal{L}_{\text{LLM}}(x_{1:\ell-1}, x_\ell) + \lambda \cdot d(\text{Pred}(\text{Enc}(x)), \text{Enc}(T_i(x)))$$

dove T_i è campionato casualmente da una famiglia di trasformazioni semanticamente invarianti. La difficoltà tecnica di questo approccio risiede esattamente nella definizione di cosa costituisca una trasformazione «semanticamente invariante»: la sinonimia lessicale, ad esempio, può modificare le implicature pragmatiche pur preservando le condizioni di verità; la parafrasi sintattica può alterare la struttura informativa (topic/focus) pur mantenendo il contenuto logico. Questa difficoltà non è un ostacolo insormontabile — è semplicemente il corrispettivo linguistico della scelta delle augmentazioni visive — ma richiede un investimento teorico specifico che LLM-JEPA non compie.

Un approccio complementare, suggerito implicitamente dall'esperimento con le parafrasi, consiste nel sfruttare modelli linguistici già addestrati per generare le viste. Se un LLM sufficientemente capace è in grado di parafrasare un testo mantenendone il significato, allora le coppie (testo originale, parafrasi generata) costituiscono viste candidate per l'addestramento JEPA. Questo crea però un circolo metodologico non banale: si utilizza un LLM per generare i dati di addestramento di un LLM-JEPA, e la qualità delle viste è limitata dalla capacità di parafrasi del modello generatore. Huang, LeCun e Balestriero non discutono questa possibilità esplicitamente, ma i risultati sull'esperimento con le parafrasi — miglioramenti statisticamente significativi su Rotten Tomatoes e Yelp dopo un solo epoch di fine-tuning su un modello preaddestrato con obiettivo JEPA su parafrasi — suggeriscono che la direzione è praticabile.

20.3 Loss dropout come leva per il preaddestramento a piena scala

Il meccanismo di loss dropout (LD) introdotto nella sezione 5.2 di LLM-JEPA rappresenta, fra tutte le proposte tecniche dei tre contributi, quella con le implicazioni operative più immediate per la scalabilità. La sua logica è elementare: se durante l'addestramento si applica il termine JEPA soltanto a una frazione $(1 - \alpha)$ dei mini-batch, scartando casualmente la perdita JEPA sul restante α, il costo computazionale per epoch si riduce da 2x (due passaggi in avanti per batch, uno per il testo e uno per il codice con la maschera personalizzata) a $(2 -$

α)x. Per α = 0.5, il costo è 1.5x quello del fine-tuning standard; per α = 0.75, si scende a 1.25x.

Gli esperimenti riportati in Tabella 6 dell'articolo mostrano che la loss dropout non soltanto riduce il costo computazionale ma può effettivamente migliorare l'accuratezza a parità di PFLOP impiegati. Questo risultato non è banale: suggerisce che l'applicazione del termine JEPA su ogni singolo batch introduce una certa ridondanza o rigidità nell'aggiornamento dei parametri, e che una regolarizzazione stocastica — analoga al dropout classico applicato ai neuroni — può avere un effetto benefico simile a quello di un tasso di apprendimento più basso o di una maggiore stochasticità nel campionamento dei batch. Gli autori notano empiricamente che «mantenere $\lambda \times (1 - \alpha)$ approssimativamente costante fornisce una linea guida utile per regolare congiuntamente λ e α al fine di bilanciare efficienza computazionale e accuratezza»: questa relazione di proporzionalità inversa fra il tasso di dropout e il peso del termine JEPA è un'euristica pratica che attende una giustificazione teorica più profonda.

In prospettiva di preaddestramento a piena scala — un LLM con decine o centinaia di miliardi di parametri addestrato su trilioni di token — l'overhead di 2x per singolo batch è economicamente proibitivo con le risorse computazionali attuali. La loss dropout con $\alpha = 0.9$ ridurrebbe l'overhead a 1.1x, rendendo LLM-JEPA potenzialmente praticabile anche a questa scala, a condizione che i miglioramenti di qualità delle rappresentazioni si mantengano statisticamente significativi. Nulla nei dati sperimentali disponibili garantisce che questo sarà il caso, ma le tendenze osservate negli esperimenti con modelli da 1B, 3B, 7B e 8B parametri — tutti riportati in Tabella 15 — mostrano miglioramenti statisticamente significativi a tutte le scale testate, il che suggerisce che il meccanismo non sia un artefatto specifico della piccola scala.

20.4 Scalabilità di Var-JEPA alla visione: il programma di Gögl e Yau come architettura concreta

La proposta di estensione di Var-JEPA alla visione abbozzata da Gögl e Yau nell'Appendice A.4 merita una discussione più dettagliata di quella offerta nel contributo originale, perché costituisce il piano architetturale più concreto disponibile per una realizzazione pratica del framework variazionale su dati visivi.

Gli autori suggeriscono quattro passi: tokenizzare immagini o video in sequenze di patch o tubelet, definire una vista context x e una vista target y mediante mascheramento, parametrizzare i due encoder variazionali con Vision Transformer (ViT), e aggiungere un decoder $p_\theta(y|s_y)$ per ottenere l'obiettivo ELBO. Il punto più delicato di questa proposta riguarda il decoder: nella versione tabulare di Var-T-JEPA, il decoder ricostruisce singole feature scalari o categoriche attraverso MLP specifiche per feature (equazione 21 dell'articolo), con una complessità computazionale proporzionale al numero di feature D_{feat}. Nella versione visiva, il decoder dovrebbe ricostruire patch di immagine — tipicamente sequenze di 196 o 256 vettori di dimensione 768 in architetture ViT-Base — da altrettante rappresentazioni latenti s_y. Il costo del decoder visivo è quindi proporzionale al numero di patch, al quadrato della loro dimensione nello spazio latente, e alla profondità del decoder scelto.

La struttura matematica dell'obiettivo ELBO rimane identica a quella derivata nel Capitolo 3 dell'articolo di Gögl e Yau:

$$\mathrm{ELBO}(x, y; \theta, \varphi) = \mathbb{E}_{q_\varphi(s_x|x)}[\log p_\theta(x|s_x)] + \mathbb{E}_{q_\varphi(s_x,z,s_y|x,y)}[\log p_\theta(y|s_y)] - \mathrm{KL}(q_\varphi(s_x|x) \,\|\, p(s_x)) - \mathrm{KL}(q_\varphi(z|s_x) \,\|\, p(z)) - \mathrm{KL}(q_\varphi(s_y|s_x, z, y) \,\|\, p_\theta(s_y|s_x, z))$$

dove ora x è una vista context dell'immagine (patch non mascherati), y è una vista target (patch mascherati), s_x e s_y sono le rappresentazioni latenti nello spazio ViT, e z è la variabile ausiliaria che cattura la variabilità di s_y non spiegabile da s_x. Le distribuzioni per le osservazioni visive — p_θ(x|s_x) e p_θ(y|s_y) — sarebbero naturalmente modellate come gaussiane sui valori dei pixel (in analogia con i VAE classici) o come distribuzioni su token discreti ottenuti da un codebook visivo preaddestrato (in analogia con i modelli di generazione di immagini basati su tokenizzatori discreti).

Il principale ostacolo pratico non è concettuale ma computazionale: il target posterior q_φ(s_y|s_x, z, y) deve elaborare la vista target completa y per parametrizzare la distribuzione variazionale, il che significa un secondo passaggio in avanti del ViT encoder su y in aggiunta a quello su x. Per immagini ad alta risoluzione o video, questo overhead può essere proibitivo. Una possibile mitigazione è applicare la loss dropout di LLM-JEPA anche a questa fase: calcolare il target posterior soltanto su una frazione α dei batch, campionando il prior p_θ(s_y|s_x, z) per i batch rimanenti. Questa ibridazione fra i due framework — non esplicitamente proposta da nessuno dei due articoli — è logicamente coerente con entrambi e potenzialmente in grado di ridurre l'overhead del target posterior senza sacrificare la qualità delle rappresentazioni, a condizione che il prior condizionato sia sufficientemente espressivo da compensare l'assenza del segnale posteriore nei batch senza dropout.

20.5 La regolarizzazione SIGReg come ponte fra Var-JEPA e LeJEPA

Un filone di ricerca immediatamente praticabile, suggerito dalla Sezione 3.7 e dall'esperimento ablativo in Tabella 1 dell'articolo di Gögl e Yau, riguarda l'integrazione sistematica della regolarizzazione SIGReg proposta in LeJEPA (Balestriero e LeCun, 2025) all'interno dell'obiettivo ELBO. La SIGReg opera sull'intera distribuzione aggregata degli embedding, misurandone la discrepanza rispetto a una gaussiana isotropa attraverso proiezioni unidimensionali casuali:

$$\mathrm{SIGReg}(A, \{e_n\}^N_{n=1}) = (1/|A|) \Sigma_{a \in A} T(\{a^\top e_n\}^N_{n=1})$$

dove A = {a_1, …, a_M} è un insieme di direzioni unitarie casuali e T misura la discrepanza fra la distribuzione empirica delle proiezioni scalari e la gaussiana standard mediante il test di Epps-Pulley o analoghi.

La relazione fra questo regolarizzatore aggregato e i termini KL per-sample dell'ELBO è chiarita dalla decomposizione ELBO-surgery di Hoffman e Johnson:

$$\mathbb{E}_x[\mathrm{KL}(q_\varphi(s_x|x) \,\|\, \mathcal{N}(0, I))] = \mathrm{KL}(q_\varphi(s_x) \,\|\, \mathcal{N}(0, I)) + I_{q_\varphi}(x; s_x)$$

Il primo termine del membro destro — KL(q_φ(s_x) ‖ $\mathcal{N}$(0, I)) — è esattamente la quantità che SIGReg approssima in modo scalabile attraverso proiezioni casuali, senza richiedere la valutazione esplicita della distribuzione aggregata q_φ(s_x). Il secondo termine — I_{q_φ}(x; s_x), l'informazione mutua fra input e latente — è un regolarizzatore di information bottleneck che penalizza la dipendenza eccessiva fra s_x e x.

La Tabella 1 di Gögl e Yau mostra che, per il latente di contesto s_x, la combinazione ELBO+SIGReg (variante D) non migliora sistematicamente rispetto al solo ELBO (variante A): le metriche diagnostiche — KL aggregata verso $\mathcal{N}$(0,I), SIGReg-MSE, ‖Cov(s) − I‖_F, ‖$\mathbb{E}$[s]‖_2 — sono pressoché identiche nelle due varianti, il che conferma empiricamente che i termini KL per-sample dell'ELBO inducono già le proprietà distributive aggregate desiderate senza richiedere SIGReg. Per il latente target s_y, invece, la situazione è diversa: poiché q_φ(s_y | s_x, z, y) è regolarizzato verso il prior condizionato p_θ(s_y | s_x, z) piuttosto che verso $\mathcal{N}$(0, I), la distribuzione aggregata q_φ(s_y) può deviare significativamente dall'isotropia gaussiana. In questo caso SIGReg applicata a s_y (variante C) riduce sensibilmente la KL aggregata — da 3.530 ± 0.377 per la variante A a 1.999 ± 0.058 per la variante C — e produce una distribuzione più isotropa senza degradare l'accuratezza del probe lineare.

Questa asimmetria — SIGReg è ridondante per s_x ma utile per s_y — ha implicazioni importanti per il design di architetture ibride che combinino JEPA variazionale e modelli linguistici generativi. Se la rappresentazione s_y deve essere condivisa fra il target posterior di Var-JEPA e il decoder generativo di LLM-JEPA, allora la sua distribuzione aggregata deve essere sia semanticamente informativa sia geometricamente regolare per permettere un efficace campionamento e decoding. SIGReg applicata selettivamente a s_y potrebbe svolgere esattamente questa funzione, agendo come un ponte fra la struttura variazionale di Var-JEPA e la struttura generativa di LLM-JEPA.

Capitolo 20 — Verso un'architettura cognitiva ibrida (continuazione)

20.6 L'incertezza latente come meccanismo di selezione attiva per l'apprendimento continuo

La capacità di Var-JEPA di produrre stime di incertezza per-sample — attraverso la varianza della distribuzione posteriore q_φ(s_y | s_x, z, y) — è presentata dagli autori principalmente come uno strumento di valutazione selettiva: scartare i campioni più incerti al momento del test e misurare l'accuratezza sul sottoinsieme rimanente. I risultati su MNIST e SIM mostrano curve rischio-copertura con area sotto la curva (AURC) di 0.123 e 0.209 rispettivamente, e curve ROC per l'identificazione di campioni ad alta ambiguità con AUC di 0.986 e 0.805 — indicatori che confermano una correlazione robusta fra incertezza latente e errore di classificazione.

Questa stessa capacità potrebbe tuttavia svolgere una funzione più attiva se inserita in un ciclo di apprendimento continuo. L'intuizione è la seguente: in un sistema che apprende

sequenzialmente su una successione di dataset D_1, D_2, …, D_T, il problema del dimenticatoio catastrofico è aggravato dal fatto che il modello non dispone di un criterio principiato per identificare quali campioni di D_t siano «nuovi» rispetto a ciò che ha appreso su D_1, …, D_{t−1} e quali siano semplicemente varianti di pattern già acquisiti. L'incertezza latente di Var-JEPA fornisce esattamente questo criterio: un campione che produce una varianza posteriore elevata — ovvero per cui la distribuzione $q_\varphi(s_y \mid s_x, z, y)$ è lontana dal prior condizionato $p_\theta(s_y \mid s_x, z)$ — è un campione che il modello corrente non riesce a rappresentare adeguatamente, e che quindi ha un alto valore informativo per l'aggiornamento dei parametri.

Formalmente, si può definire un punteggio di novità u_i per ciascun campione (x_i, y_i) come:

$$u_i = (1/D_{\text{feat}}) \Sigma_j \mathrm{KL}(q_\varphi(s^{(j)}_{y_k} \mid s_x, z, y; m^{(k)}_{\text{trg}}) \,\|\, p_\theta(s^{(j)}_{y_k} \mid s_x, z; m^{(k)}_{\text{trg}}))$$

che è esattamente il termine $\mathcal{L}^{\mathrm{KL}}_{s_y}$ della perdita ELBO calcolato per il singolo campione. I campioni con u_i elevato sono quelli per i quali il posterior target si allontana maggiormente dal prior predittivo, indicando che il modello corrente fatica a predire le rappresentazioni target dal contesto. Una strategia di selezione del buffer di replay per l'apprendimento continuo — analoga alle tecniche di experience replay in uso nell'apprendimento per rinforzo — potrebbe prioritizzare i campioni con u_i elevato, conservando in memoria quelli che meglio rappresentano le zone di incertezza del modello.

Gögl e Yau non sviluppano questa direzione nell'articolo, e sarebbe scorretto presentarla come una proposta compiuta piuttosto che come un'ipotesi di ricerca derivata dalla loro struttura. Tuttavia, la coerenza matematica di questa estensione con il framework esistente è sufficiente per considerarla una delle direzioni future più naturali e meglio motivate dal punto di vista teorico.

20.7 Verso un obiettivo unificato: LLM-Var-JEPA

La domanda che emerge con maggiore urgenza dal confronto fra LLM-JEPA e Var-JEPA è se i due framework possano essere integrati in un obiettivo unificato che sia simultaneamente generativo (capace di completare testi e generare risposte coerenti), variazionale (capace di quantificare l'incertezza sulla propria rappresentazione latente) e predittivo nello spazio degli embedding (capace di allineare le rappresentazioni di viste complementari dello stesso contenuto).

La struttura di un tale obiettivo può essere abbozzata partendo dall'ELBO di Var-JEPA e sostituendo i termini di ricostruzione e generazione con perdite autoregressivecross-entropy. Il termine $\mathcal{L}_{\text{gen}}$ — che nella Var-JEPA tabulare misura il log-likelihood negativo di y dato il latente s_y — diventerebbe in questa versione linguistica:

$$\mathcal{L}^{\mathrm{LM}}_{\text{gen}} = -\,\mathbb{E}_{q_\varphi(s_x, z, s_y \mid x, y)} \left[\Sigma_{\ell=1}^{|y|} \log p_\theta(y_\ell \mid y_{1:\ell-1}, s_y) \right]$$

ovvero la cross-entropy autoregressiva condizionata sul latente target s_y. Analogamente, il termine $\mathcal{L}_{rec}$ — che ricostruisce il contesto x dal latente s_x — diventerebbe:

$$\mathcal{L}_{rec}^{LM} = - \mathbb{E}_{q_\varphi(s_x|x)} [\Sigma_{\ell=1}^{|x|} \log p_\theta(x_\ell \mid x_{1:\ell-1}, s_x)]$$

L'obiettivo completo del sistema ipotetico LLM-Var-JEPA sarebbe quindi:

$$\mathcal{L}_{LLM\text{-}Var\text{-}JEPA} = \alpha^{rec} \cdot \mathcal{L}_{rec}^{LM} + \alpha^{gen} \cdot \mathcal{L}_{gen}^{LM} + \alpha^{KL}_{s_x} \cdot KL(q_\varphi(s_x|x) \| \mathcal{N}(0, I)) + \alpha^{KL}_{z} \cdot KL(q_\varphi(z|s_x) \| \mathcal{N}(0, I)) + \alpha^{KL}_{s_y} \cdot KL(q_\varphi(s_y|s_x, z, y) \| p_\theta(s_y|s_x, z))$$

Il termine $KL(q_\varphi(s_y|s_x, z, y) \| p_\theta(s_y|s_x, z))$ svolge esattamente la funzione del termine JEPA in LLM-JEPA — allineare la rappresentazione del target con la predizione del contesto — ma in modo probabilistico piuttosto che deterministico: invece di minimizzare la distanza coseno fra Pred(Enc(Text)) e Enc(Code), minimizza la KL-divergenza fra la distribuzione posteriore del target e il prior condizionato predittivo. Questo ha due vantaggi rispetto alla versione deterministica: il training è più stabile perché la KL penalizza la divergenza in entrambe le direzioni (sia la media sia la varianza delle distribuzioni), e la varianza posteriore fornisce un segnale di incertezza sulla qualità della predizione, come discusso nel paragrafo precedente.

Il costo computazionale aggiuntivo rispetto al solo fine-tuning standard sarebbe sostanzialmente equivalente a quello di LLM-JEPA con target posterior — cioè il costo di un passaggio aggiuntivo del transformer — ma con il beneficio aggiuntivo dell'incertezza latente e della struttura variazionale. La loss dropout di LLM-JEPA potrebbe essere applicata selettivamente ai termini più costosi — in particolare al target posterior $q_\varphi(s_y|s_x, z, y)$ — mantenendo comunque il gradiente attraverso i termini $\mathcal{L}_{rec}^{LM}$ e $\mathcal{L}_{gen}^{LM}$ su tutti i batch.

Va ribadito che questa formulazione non è proposta in alcuno dei tre articoli: è una sintesi logicamente motivata dalle strutture matematiche che essi presentano separatamente. La sua fattibilità empirica rimane da verificare.

20.8 JEPA come componente di un'architettura cognitiva integrata: la visione di Mumuni e Mumuni

Il contributo di Mumuni e Mumuni propone nella sezione 7 un modello concettuale di sistema AGI che unifica incarnazione, radicamento simbolico, causalità e memoria in un'architettura a blocchi funzionali. In questa architettura, il blocco di incarnazione fornisce i meccanismi di percezione e azione sul mondo; il blocco di radicamento simbolico connette le rappresentazioni astratte dell'LLM alle entità fisiche corrispondenti; il blocco causale apprende le relazioni causa-effetto attraverso l'interazione con il mondo; il blocco mnemonico conserva, organizza e rende accessibili le conoscenze acquisite.

JEPA — nella sua versione linguistica o variazionale — non è esplicitamente menzionato da Mumuni e Mumuni come componente di questa architettura: i due articoli provengono da tradizioni di ricerca distinte e non si citano a vicenda. Tuttavia, il ruolo che JEPA potrebbe svolgere all'interno del modello di Mumuni e Mumuni è identificabile con precisione: il

sistema di predizione nello spazio degli embedding corrisponderebbe al blocco di elaborazione delle rappresentazioni percettive, producendo le codifiche latenti che il sistema di radicamento utilizzerebbe per connettere simboli e referenti, e che il sistema causale utilizzerebbe come variabili su cui costruire i propri modelli strutturali.

Concretamente, si può immaginare la seguente articolazione. Il modello JEPA — sia esso LLM-JEPA, Var-JEPA o un futuro LLM-Var-JEPA — genera per ciascuna osservazione (x, y) una coppia di rappresentazioni latenti (s_x, s_y) con le proprietà geometriche desiderate: isotropia approssimativa per s_x e z, struttura predittiva per il prior condizionato p_θ(s_y|s_x, z), informazione di incertezza attraverso la varianza posteriore. Queste rappresentazioni costituiscono l'input per un sistema di radicamento simbolico che apprende a connettere le coordinate latenti a nodi in un grafo di conoscenza: la coppia (s_x, s_y) viene interpretata come due proiezioni dello stesso nodo semantico, e la relazione fra le loro coordinate latenti viene utilizzata per inferire e aggiornare i legami nel grafo. Il sistema causale, a sua volta, opera sulle rappresentazioni latenti per apprendere relazioni di tipo interventistico: se una perturbazione di s_x (ottenuta, ad esempio, modificando una feature semantica del contesto di input) produce una variazione sistematica e prevedibile di s_y (rilevabile attraverso il prior condizionato p_θ(s_y|s_x, z)), allora si può inferire una relazione causale fra i due aspetti del contenuto rappresentato.

Questa integrazione è programmaticamente coerente con la visione di Mumuni e Mumuni, che scrivono che «il progresso verso l'AGI richiederà che questi concetti siano trattati come un insieme unificato di primitive interrelate e complementari che modellano congiuntamente gli agenti intelligenti e il loro ambiente». Il contributo specifico di JEPA in questo quadro sarebbe quello di fornire le rappresentazioni latenti ricche e strutturate su cui gli altri moduli possono operare, senza sostituire tali moduli ma abilitandoli.

20.9 Il radicamento come vincolo sulla struttura latente: un'ipotesi derivabile dai framework esistenti

Il paragrafo precedente ha delineato un'integrazione architetturale a livello di sistema; questo paragrafo si concentra su un possibile vincolo matematico che potrebbe avvicinare il radicamento simbolico alla struttura dell'ELBO. L'intuizione di partenza è che il radicamento nel senso di Harnad richiede che le rappresentazioni latenti siano sufficientemente informative sulle proprietà percepibili degli oggetti e dei fenomeni cui si riferiscono: un simbolo è radicato se e solo se la sua rappresentazione codifica in modo accessibile le caratteristiche sensorimotorie del proprio referente.

In un sistema JEPA applicato a coppie (testo, percezione) — dove il testo descrive un oggetto e la percezione è un'immagine o una sequenza di letture sensoriali dell'oggetto stesso — il termine $\mathcal{L}_{rec}^{LM}$ che ricostruisce il testo dal latente s_x fornisce un vincolo di fedeltà testuale; il termine $\mathcal{L}_{gen}$ che ricostruisce la percezione y dal latente s_y fornisce un vincolo di fedeltà percettiva; e il termine $KL(q_\varphi(s_y|s_x, z, y) \| p_\theta(s_y|s_x, z))$ fornisce un vincolo di allineamento fra le due modalità. Insieme, questi tre vincoli impongono che s_x e s_y siano rappresentazioni dello stesso contenuto semantico espresso in due modalità diverse, e che il prior condizionato $p_\theta(s_y|s_x, z)$ apprenda a predire la rappresentazione percettiva dalla rappresentazione testuale.

Questo non è ancora radicamento nel senso pieno del termine — non garantisce che le coordinate latenti corrispondano a proprietà fisiche specifiche e interpretabili — ma è una forma di allineamento intermodale che soddisfa almeno la condizione necessaria: le rappresentazioni testuali devono contenere abbastanza informazione sulle proprietà percettive degli oggetti da permettere la predizione delle rappresentazioni visive corrispondenti. La distanza fra questo allineamento intermodale e il radicamento propriamente detto è quella fra la correlazione statistica e la connessione semantica: due sistemi possono essere correlati senza che l'uno esplichi il significato dell'altro. Tuttavia, l'allineamento intermodale è una condizione necessaria (sebbene non sufficiente) per il radicamento, e i framework JEPA forniscono un meccanismo principiato per ottenerlo.

20.10 La valutazione dell'architettura ibrida: criteri e protocolli

Qualunque proposta di architettura cognitiva ibrida che integri JEPA, radicamento simbolico, causalità e memoria deve confrontarsi con il problema valutativo che Mumuni e Mumuni riconoscono esplicitamente come irrisolto: «valutare e accertare quando l'intelligenza generale venga raggiunta è un problema difficile». Le metriche disponibili nei tre contributi — accuratezza su benchmark di classification e traduzione per LLM-JEPA, accuratezza a valle con probe lineare e curve rischio-copertura per Var-JEPA — misurano aspetti specifici e parziali della qualità delle rappresentazioni, non la generalità o la robustezza cognitiva del sistema nel suo complesso.

Un protocollo di valutazione più appropriato per un'architettura ibrida dovrebbe includere almeno tre dimensioni distinte. La prima è la trasferibilità zero-shot: quanto bene le rappresentazioni apprese su un insieme di compiti si trasferiscono a compiti nuovi, con distribuzione di input diversa da quella di addestramento, senza aggiornamento dei parametri. La seconda è la robustezza agli interventi: quanto le previsioni del sistema cambiano in modo coerente quando il contesto di input viene sistematicamente modificato in modo semanticamente rilevante (ad esempio, invertendo la polarità di un'affermazione causale). La terza è la calibrazione dell'incertezza: quanto la varianza posteriore del modello variazionale correla con la probabilità di errore su campioni non visti, includendo campioni provenienti da distribuzioni diverse da quella di addestramento.

Var-JEPA fornisce le basi per la terza dimensione — le figure 3 dell'articolo mostrano curve rischio-copertura e ROC che attestano una correlazione robusta fra incertezza latente e errore su MNIST e SIM — ma la valutazione si limita a distribuzioni relativamente semplici con rumore controllato. LLM-JEPA fornisce elementi per la prima dimensione — i risultati di preaddestramento su cestwc/paraphrase con successivo fine-tuning su Rotten Tomatoes e Yelp mostrano che i benefici del preaddestramento JEPA si trasferiscono a task diversi — ma le distribuzioni valutate rimangono vicine a quella di preaddestramento. La seconda dimensione — robustezza agli interventi, che è l'aspetto più direttamente legato alla causalità — non è misurata sistematicamente da nessuno dei tre contributi.

20.11 La distanza che rimane: un bilancio onesto

Il presente capitolo ha articolato un insieme di ipotesi di ricerca future che emergono direttamente dalle strutture matematiche e dalle ammissioni programmatiche dei tre contributi esaminati. Alcune di queste ipotesi — la loss dropout per il preaddestramento a piena scala di LLM-JEPA, la costruzione di viste mediante parafrasi automatiche, l'applicazione di SIGReg al latente target in una versione ibrida ELBO+SIGReg — sono estensioni tecniche relativamente contenute, praticabili con risorse computazionali moderate e verificabili empiricamente nel breve periodo. Altre — l'integrazione di JEPA in un'architettura cognitiva che includa incarnazione, radicamento, causalità e memoria; la costruzione di un obiettivo LLM-Var-JEPA unificato; l'uso dell'incertezza latente come meccanismo di selezione attiva per l'apprendimento continuo — sono programmi di ricerca di medio-lungo periodo, il cui successo dipende da sviluppi teorici ed empirici che i tre contributi analizzati rendono plausibili ma non garantiscono.

Ciò che i tre lavori non suggeriscono — e che sarebbe disonesto presentare come un'ipotesi di ricerca motivata dal testo — è qualunque percorso diretto verso la risoluzione del problema causale a livello fondazionale. La distinzione fra distribuzione osservazionale e distribuzione interventistica, che è il cuore del problema causale, non può essere affrontata modificando la funzione di perdita di un sistema di apprendimento per rappresentazione: richiede o l'incorporazione esplicita di assunzioni strutturali sul processo generativo dei dati (nella forma di modelli causali strutturali), o l'interazione con ambienti in cui gli interventi sono realizzabili e le loro conseguenze osservabili. Questa seconda strada — che Mumuni e Mumuni associano all'incarnazione e all'apprendimento attivo — rimane la più coerente con il paradigma cognitivo che i tre contributi, ciascuno a proprio modo, cercano di avvicinare.

La distanza fra il punto in cui ci troviamo — modelli linguistici che apprendono rappresentazioni più strutturate attraverso obiettivi JEPA, sistemi variazionali che quantificano l'incertezza nello spazio latente, architetture concettuali che identificano le componenti mancanti per l'intelligenza generale — e il punto verso cui ci dirigiamo — sistemi cognitivi capaci di radicamento simbolico robusto, ragionamento causale genuino, apprendimento continuo senza dimenticatoio, e interazione incarnata con il mondo fisico — è una distanza che i tre contributi contribuiscono a misurare con precisione crescente. Non si tratta di un progresso modesto: sapere esattamente quanto manca è una forma di conoscenza indispensabile per non sprecare le risorse intellettuali e computazionali disponibili su direzioni che non conducono alla destinazione cercata. È, in un senso preciso, il tipo di progresso che la filosofia della scienza riconosce come costitutivo delle fasi di transizione paradigmatica: non ancora il nuovo paradigma, ma la mappa dettagliata del terreno che separa il vecchio dal nuovo.

20.12 Il rischio del "lavaggio epistemico" e la graduazione della plausibilità

[Integrazione da Rapp-JEPA, Cap. X–XI] Il bilancio di §20.11 conclude onestamente che "sapere esattamente quanto manca è una forma di conoscenza indispensabile". A questo si deve aggiungere un avvertimento metodologico che Rapp-JEPA chiama meccanismo del "lavaggio epistemico": il processo per cui il successo empirico su benchmark sempre più sofisticati genera l'impressione che le limitazioni fondamentali siano state superate,

quando in realtà il problema è stato ridefinito in modo da aggirarle. Per i tre framework analizzati in questa monografia, il rischio si manifesta in modo specifico: l'aumento di accuracy su NL-RX-SYNTH, GSM8K e Spider documenta un miglioramento reale nella qualità delle rappresentazioni, ma non costituisce evidenza che il sistema si stia avvicinando a forme di comprensione causalmente radicata; la calibrazione dell'incertezza latente di Var-T-JEPA su MNIST e SIM misura una correlazione robusta fra varianza posteriore ed errore, ma non è una misura di radicamento simbolico; la resistenza all'overfitting indotta dall'obiettivo JEPA riflette una proprietà geometrica delle rappresentazioni, non una proprietà semantica.

Sulla base dell'analisi sviluppata in questa monografia e in Rapp-JEPA (§X.4), è possibile formulare un giudizio differenziato sulla plausibilità della via verso l'AGI articolato su quattro livelli. La plausibilità è alta per i componenti individuali: Var-T-JEPA, LLM-JEPA e Var-JEPA sono architetture funzionanti con proprietà teoriche dimostrate e risultati empirici significativi. La plausibilità è media per l'integrazione dei componenti: il framework LLM-Var-JEPA unificato proposto in §20.7 è architetturalmente coerente ma non è mai stato costruito, e l'addestramento congiunto di sistemi variazionali e autoregressivi a piena scala presenta difficoltà tecniche non ancora risolte. La plausibilità è bassa per la risoluzione delle barriere fondamentali nell'orizzonte di un decennio: la generalizzazione composizionale, il radicamento causale nell'embodiment e il catastrophic forgetting sono problemi di ricerca aperta che hanno resistito a soluzioni definitive per decenni. La plausibilità è indeterminata per il salto qualitativo dalla competenza funzionale alla comprensione strutturalmente radicata: come i tre teoremi di impossibilità discussi in §19.13 chiariscono, non esiste attualmente un framework teorico che permetta di misurare questa distanza.

Questo giudizio differenziato non è pessimismo: è la condizione epistemica necessaria per un programma di ricerca rigoroso. I contributi analizzati — Var-JEPA, Var-T-JEPA, LLM-JEPA — sono passi autentici e significativi in una direzione teoricamente fondata. Il framework JEPA→AGI resta, nella migliore delle ipotesi, una via plausibile piuttosto che una via dimostrata. Trattarla come tale — con rigore tecnico nelle deduzioni formali e cautela nelle affermazioni prognostiche — è la forma di onestà intellettuale che la complessità del problema richiede.

20.13 Agenda di ricerca a breve termine: esperimenti prioritari con criteri di successo definiti

Per rendere operativo il programma di ricerca delineato nei capitoli precedenti, questa sezione identifica tre esperimenti prioritari realizzabili nel breve termine (12-24 mesi) con risorse standard, specificando per ciascuno design, metriche e criteri di successo.

Il primo esperimento prioritario è la validazione di Var-T-JEPA su dataset ad alta dimensionalità ($D_{feat} > 500$). Si confrontano: (a) Var-T-JEPA standard con full attention, (b) Var-T-JEPA con linear attention nel target posterior, (c) Var-T-JEPA con sparse attention basata su correlazione empirica. Metrica: accuratezza downstream vs. costo computazionale (PFLOPs). Criterio di successo: la variante con linear attention

deve raggiungere ≥ 95% dell'accuratezza full-attention con ≤ 50% del costo. Il secondo è la costruzione automatica di viste linguistiche: addestrare un modello di paraphrase generation, generare 1M di viste sintetiche da Common Crawl, addestrare LLM-JEPA e valutare su NL-RX-SYNTH e GSM8K. Criterio: LLM-JEPA con viste sintetiche deve superare il baseline senza JEPA su entrambi i dataset.

Il terzo è la decomposizione epistemica/aleatoria dell'incertezza in Var-T-JEPA: addestrare K=5 modelli con diverse inizializzazioni, calcolare incertezza aleatoria (media delle varianze) e epistemica (varianza delle medie), verificare la correlazione con difficoltà simulata. Criterio: l'incertezza epistemica deve correlare più altamente con la difficoltà OOD rispetto all'incertezza aleatoria. Questi tre esperimenti, insieme, produrrebbero dati empirici sufficienti per validare o falsificare le ipotesi teoriche centrali di questa monografia.

20.14 Visione a lungo termine: architettura cognitiva ibrida e convergenza verso l'intelligenza incarnata

La visione a lungo termine del programma JEPA converge verso un'architettura cognitiva ibrida a quattro livelli: (1) percezione multimodale variazionale attraverso encoder gaussiani per ogni modalità, (2) predizione cross-modale attraverso un prior condizionale multimodale appreso — il world model di LeCun in veste variazionale, (3) ragionamento causale attraverso un grafo causale sui latenti modali appreso dai dati, (4) pianificazione attraverso Model Predictive Control nello spazio latente. La realizzazione di questa visione richiede progressi simultanei su tutti quattro i livelli, con un grado di integrazione che va al di là di qualunque contributo singolo. Il valore della visione non è nella sua immediata realizzabilità ma nella funzione di bussola che fornisce per la ricerca a breve termine: ogni esperimento prioritario è un passo verso uno dei quattro livelli; ogni limite strutturale identificato è una barriera che il framework deve eventualmente superare.

Il framework JEPA variazionale — da Var-JEPA a Var-T-JEPA a LLM-Var-JEPA — è il contributo specifico di questa linea di ricerca verso la realizzazione di questa architettura. Non è il traguardo, ma un passo avanzato e principiato in una direzione teoricamente fondata. La coerenza matematica tra le strutture probabilistiche di Var-JEPA, la quasi-linearità dello spazio latente di LLM-JEPA, e i principi cognitivi di Mumuni e Mumuni non è accidentale: riflette una convergenza strutturale tra la teoria dell'inferenza variazionale, l'architettura dei transformer, e la psicologia cognitiva computazionale che costituisce il nucleo teorico più solido del programma di ricerca verso l'AGI.

Appendice A — Derivazione Formale dell'ELBO di Var-JEPA

La presente appendice dettaglia i passaggi algebrici completi che conducono dalla log-verosimiglianza marginale del modello generativo all'Evidence Lower Bound (ELBO) a cinque termini descritto nel Capitolo 3. La derivazione segue Gögl e Yau (2026) e si appoggia sul teorema di Bayes e sulla disuguaglianza di Jensen applicata alla funzione logaritmo concava.

A.1 Il punto di partenza: log-verosimiglianza intrattatabile

Sia il processo generativo fattorizzato come: $p_\theta(x, y, s_x, z, s_y) = p(s_x) \cdot p(z) \cdot p_\theta(x|s_x) \cdot p_\theta(s_y|s_x, z) \cdot p_\theta(y|s_y)$. La log-verosimiglianza marginale è:

$$\log p_\theta(x, y) = \log \iiint p_\theta(x, y, s_x, z, s_y)\, ds_x\, dz\, ds_y$$

Questo integrale triplo è intrattatabile per via delle non-linearità neurali nelle reti generative. Si introduce pertanto una distribuzione variazionale $q_\varphi(s_x, z, s_y \mid x, y)$ e si moltiplica e divide l'integrando per essa.

A.2 Applicazione della disuguaglianza di Jensen

Partendo da:

$$\log p_\theta(x, y) = \log \mathbb{E}_{q_\varphi}\left[p_\theta(x, y, s_x, z, s_y) / q_\varphi(s_x, z, s_y|x, y) \right]$$

La concavità del logaritmo e la disuguaglianza di Jensen ($\log \mathbb{E}[f] \geq \mathbb{E}[\log f]$ per $f \geq 0$) forniscono:

$$\log p_\theta(x, y) \geq \mathbb{E}_{q_\varphi}\left[\log p_\theta(x, y, s_x, z, s_y) - \log q_\varphi(s_x, z, s_y|x, y) \right] \equiv \mathrm{ELBO}(x, y; \theta, \varphi)$$

Il divario fra la log-verosimiglianza e l'ELBO è esattamente $\mathrm{KL}(q_\varphi \,\|\, p_\theta(\cdot|x, y)) \geq 0$, con uguaglianza se e solo se q_φ coincide con il posteriore esatto.

A.3 Sostituzione delle fattorizzazioni

Si sostituisce la fattorizzazione del modello generativo e quella del posteriore variazionale $q_\varphi(s_x, z, s_y|x, y) = q_\varphi(s_x|x) \cdot q_\varphi(z|s_x) \cdot q_\varphi(s_y|s_x, z, y)$. Si ottiene:

$$\begin{aligned}
\mathrm{ELBO} = {} & \mathbb{E}_{q_\varphi(s_x|x)}\left[\log p_\theta(x|s_x)\right] \quad \text{[termine A: ricostruzione del contesto]} \\
& + \mathbb{E}_{q_\varphi(s_x,z,s_y|x,y)}\left[\log p_\theta(y|s_y)\right] \quad \text{[termine B: generazione del target]} \\
& + \mathbb{E}_{q_\varphi(s_x|x)}\left[\log p(s_x) - \log q_\varphi(s_x|x)\right] \quad \text{[termine C]} \\
& + \mathbb{E}_{q_\varphi(sx,z|x)}\left[\log p(z) - \log q_\varphi(z|s_x)\right] \quad \text{[termine D]} \\
& + \mathbb{E}_{q_\varphi(s_x,z,s_y|x,y)}\left[\log p_\theta(s_y|s_x,z) - \log q_\varphi(s_y|s_x,z,y)\right] \quad \text{[termine E]}
\end{aligned}$$

I termini C, D ed E si riconoscono come meno la KL-divergenza corrispondente, per l'identità $\mathbb{E}_q[\log p - \log q] = -\mathrm{KL}(q \,\|\, p)$. Si ottiene così la forma finale a cinque termini.

A.4 Forma chiusa delle KL per distribuzioni gaussiane diagonali

Per la coppia di gaussiane diagonali $q = \mathcal{N}(\mu_q, \mathrm{diag}(\sigma^2_q))$ e $p = \mathcal{N}(\mu_p, \mathrm{diag}(\sigma^2_p))$, la KL-divergenza si calcola in forma chiusa:

$$KL(q \,\|\, p) = \tfrac{1}{2} \Sigma_d \left[\sigma^2_{q,d}/\sigma^2_{p,d} + (\mu_{p,d} - \mu_{q,d})^2/\sigma^2_{p,d} + \log(\sigma^2_{p,d}/\sigma^2_{q,d}) - 1 \right]$$

Nel caso speciale $p = \mathcal{N}(0, I)$ — che si applica ai termini KL per s_x e z — la formula si semplifica a:

$$KL(\mathcal{N}(\mu, \mathrm{diag}(\sigma^2)) \,\|\, \mathcal{N}(0, I)) = \tfrac{1}{2} \Sigma_d \left[\sigma^2_d + \mu^2_d - \log \sigma^2_d - 1 \right]$$

Questa espressione è analitica, priva di campionamento stocastico, e differenziabile rispetto a μ e σ^2 tramite retropropagazione automatica.

A.5 Il reparametrization trick per il campionamento differenziabile

Per i termini di ricostruzione $\mathcal{L}_{rec}$ e generazione $\mathcal{L}_{gen}$, la stima Monte Carlo richiede campionamento da distribuzioni gaussiane dipendenti dai parametri φ. Il reparametrization trick di Kingma e Welling (2013) esprime il campione come:

$$s_x = \mu^{s_x}_\varphi(x) + \sigma^{s_x}_\varphi(x) \odot \varepsilon_{s_x}, \quad \varepsilon_{s_x} \sim \mathcal{N}(0, I)$$

dove $\odot$ è il prodotto elemento per elemento. Il campione è ora una funzione deterministica di (x, ε), con ε trattato come costante rispetto alla differenziazione. Stima a singolo campione per mini-batch è sufficiente per la convergenza pratica.

A.6 JEPA deterministico come caso limite

Quando le varianze dei posteriori tendono a zero — $\sigma^{2\wedge}\{s_x\}_\varphi \to 0$, $\sigma^{2\wedge}z_\varphi \to 0$, $\sigma^{2\wedge}\{s_y\}_\varphi \to 0$ — le distribuzioni degenerano in delta di Dirac centrate sulle medie. Il termine KL su s_y diventa:

$$KL(q_\varphi(s_y \mid s_x,z,y) \,\|\, p_\theta(s_y \mid s_x,z)) \to (1/2\sigma^2_{prior}) \cdot \|\mu^{s_y}_\varphi(s_x,z,y) - \mu^{s_y}_\theta(s_x,z)\|^2_2$$

Identificando $\mu^{s_y}_\varphi \to f_{trg}(y)$ e $\mu^{s_y}_\theta \to g(s_x,z)$, questo termine converge a $(1/2\sigma^2_{prior}) \cdot \|s_y - \hat{s}_y\|^2_2$ — esattamente la perdita JEPA canonica, a meno di una costante di normalizzazione. Il JEPA deterministico è dunque il caso limite $\sigma \to 0$ dell'ELBO variazionale.

Appendice B — Dettagli Implementativi di Var-T-JEPA

La presente appendice riporta le specifiche architetturali e i dettagli di implementazione di Var-T-JEPA per dati tabulari, seguendo Gögl e Yau (2026). Si documenta la tokenizzazione, le reti variazionali, la strategia di mascheramento, il KL annealing e gli iperparametri utilizzati nelle valutazioni empiriche.

B.1 Tokenizzazione delle feature eterogenee

Per le variabili numeriche $j \in I_{num}$, il token risultante è:

$$t^{(j)} = W_{num} \cdot x^{(j)} + b^{(j)} + e^{num}_{type} + e^{(j)}_{pos}$$

dove $W_{num} \in \mathbb{R}^{D_{hid} \times 1}$ è la proiezione lineare condivisa, $b^{(j)} \in \mathbb{R}^{D_{hid}}$ il bias specifico della feature, e^{num}_{type} l'embedding di tipo numerico, $e^{(j)}_{pos}$ l'embedding posizionale per la colonna j. Per le variabili categoriali $j \in I_{cat}$ con L_j categorie:

$$t^{(j)} = \text{Embed}_{L_j}(x^{(j)}) + e^{cat}_{type} + e^{(j)}_{pos}$$

con tabelle di embedding $\text{Embed}_{L_j} : \{0,\ldots,L_j-1\} \to \mathbb{R}^{D_{hid}}$ apprese per ciascuna variabile.

B.2 Strategia di mascheramento

Le dimensioni delle maschere sono campionate uniformemente per ogni batch:

$$M_{ctx} \sim \text{Uniform}[\lfloor D_{feat} \cdot r^{min}_{ctx} \rfloor, \lfloor D_{feat} \cdot r^{max}_{ctx} \rfloor]$$

$$M_{trg} \sim \text{Uniform}[\lfloor D_{feat} \cdot r^{min}_{trg} \rfloor, \lfloor D_{feat} \cdot r^{max}_{trg} \rfloor]$$

con il vincolo $M_{ctx} + M_{trg} \leq D_{feat}$ per garantire la non-sovrapposizione. Per ogni campione si genera una vista di contesto $x = m_{ctx} \odot w$ e K viste target $y_k = m^{(k)}_{trg} \odot w$, $k = 1,\ldots,K$.

B.3 Architettura dei sei moduli variazionali

I sei moduli che implementano Var-T-JEPA sono i seguenti.

(1) Context Encoder $q_\varphi(s^{(j)}_x \mid x)$: Transformer con preposti N_{CLS} token CLS, seguiti da proiezioni lineari per media e log-varianza di ciascun token latente.

(2) Auxiliary Encoder $q_\varphi(z \mid s_x)$: MLP compatto su rappresentazioni di contesto aggregate via attention pooling su L_{pool} token: $\bar{s}_x = \text{AttentionPool}(s_x)$. Output: μ^z_φ, $\log \sigma^z_\varphi$.

(3) Target Posterior $q_\varphi(s^{(j)}_{y_k} \mid s_x, z, w; m^{(k)}_{trg})$: Transformer che concatena token CLS, token pooled del contesto $t^{(i)}_{\bar{s}_x} = W_{sx} \cdot \bar{s}^{(i)}_x + e^{pooled\text{-}ctx}_{type}$, token ausiliario $t_z = W_z \cdot z + e^{aux\text{-}latent}_{type}$, e token di tutte le feature w. Produce μ^{sw}_φ e Σ^{sw}_φ per le feature mascherate.

(4) Predictive Model p_θ(s^{(j)}_{y_k} | s_x, z; m^{(k)}_trg): Transformer con token di contesto h_ctx = Linear(s_x) + Linear(z) + e^{ctx}_pos e token maschera apprendibili h_mask = MaskToken + Linear(z) + e^{trg}_pos. Produce μ^{sy}_θ e Σ^{sy}_θ.

(5) Context Decoder p_θ(x^{(j)} | s^{(j)}_x): MLP specifico per feature, che mappa s^{(j)}_x → $\mathbb{R}^1$ (numerico) o → $\mathbb{R}^{C_j}$ (categoriale con C_j classi).

(6) Target Decoder p_θ(y^{(j)} | s^{(j)}_y): stessa architettura del context decoder applicata a s^{(j)}_w.

B.4 Termini di perdita per dati tabulari

Adattando l'ELBO generale al mascheramento per feature, i cinque termini per Var-T-JEPA sono:

$\mathcal{L}$_rec = (1/M_ctx) Σ_{j∈m_ctx} [−log p_θ(x^{(j)} | s^{(j)}_x)]

$\mathcal{L}$_gen = (1/D_feat) Σ^{D_feat}_{j=1} [−log p_θ(w^{(j)} | s^{(j)}_w)]

$\mathcal{L}$^{KL}_{s_x} = (1/M_ctx) Σ_{j∈m_ctx} KL(q_φ(s^{(j)}_x | x) ‖ $\mathcal{N}$(0,I))

$\mathcal{L}$^{KL}_z = KL(q_φ(z | s_x) ‖ $\mathcal{N}$(0,I))

$\mathcal{L}$^{KL}_{s_y} = (1/(K·M_trg)) Σ^K_{k=1} Σ_{j∈m^{(k)}_trg} KL(q_φ(s^{(j)}_{y_k} | s_x,z,w;m^{(k)}_trg) ‖ p_θ(s^{(j)}_{y_k} | s_x,z;m^{(k)}_trg))

La perdita totale è: $\mathcal{L}$ = α^{rec}·$\mathcal{L}$_rec + α^{gen}·$\mathcal{L}$_gen + α^{KL}_{s_x}(t)·$\mathcal{L}$^{KL}_{s_x} + α^{KL}_z(t)·$\mathcal{L}$^{KL}_z + α^{KL}_{s_y}(t)·$\mathcal{L}$^{KL}_{s_y}

B.5 KL Annealing

I pesi dei termini KL seguono schedulations di annealing lineari indipendenti:

α^{KL}_·(t) = min(t / T^{anneal}_·, 1) · α^{KL}_{·,final}

dove t è il passo corrente e T^{anneal}_· è la durata dell'annealing per ciascun termine. Questa tecnica previene il collasso posteriore nelle fasi iniziali dell'ottimizzazione, consentendo al modello di sviluppare prima buone rappresentazioni ricostruttive.

B.6 Iperparametri per le valutazioni downstream

La seguente tabella riporta gli iperparametri principali di Var-T-JEPA per i dataset utilizzati nelle valutazioni empiriche. Le notazioni sono: batch = dimensione del batch; lr = learning rate; d = dimensione nascosta; L = numero di strati; H = numero di teste di attenzione; ff = dimensione feedforward; Lp = dimensione dell'embedding del predictor; K = numero di predizioni target per campione.

Adult (AD): batch=512, lr=0.001, d=64, L=8, H=4, ff=256, K=4, α^{KL}_{sx}=1e-4, α^{KL}_{sy}=1e-5, α^{rec}=0.1, α^{gen}=1.

Covertype (CO): batch=512, lr=5e-4, d=64, L=8, H=8, ff=64, K=4, α^{KL}_{sx}=1e-6, α^{KL}_{sy}=1e-6, α^{rec}=0.001, α^{gen}=0.5.

Electricity (EL): batch=512, lr=5e-4, d=64, L=6, H=4, ff=64, K=4, α^{KL}_{sx}=1e-6, α^{KL}_{sy}=1e-6, α^{rec}=0.001, α^{gen}=0.25.

Credit Card (CC) e Bank Marketing (BM): batch=512, lr=0.001, d=64, L=8, H=4, ff=256, K=4, stesso schema di Adult.

MNIST: batch=256, lr=0.001, d=64, L=8, H=4, ff=128, K=4. SIM: batch=512, lr=5e-4, d=64, L=16, H=2, ff=64, K=4.

Appendice C — Dettagli Sperimentali di LLM-JEPA

La presente appendice documenta il protocollo sperimentale, le architetture del predittore, la maschera di attenzione personalizzata e i risultati supplementari di LLM-JEPA secondo Huang, LeCun e Balestriero (2025).

C.1 Implementazione della maschera di attenzione a blocchi causali

Per ottenere Enc(Text) ed Enc(Code) in modo indipendente in un unico passaggio in avanti, si costruisce una maschera di attenzione additiva M ∈ ℝ^{(t_size+c_size)×(t_size+c_size)} inizializzata a $-\infty$ e con maschere triangolari inferiori inserite per ciascun blocco:

M[i, j] = 0 se (i,j) ∈ blocco_Text × blocco_Text e i ≥ j

M[i, j] = 0 se (i,j) ∈ blocco_Code × blocco_Code e i ≥ j

M[i, j] = $-\infty$ altrimenti

In questo modo l'attenzione è causale all'interno di ciascun blocco ma completamente bloccata tra i blocchi. L'implementazione in PyTorch sfrutta il meccanismo di additive attention mask dei modelli HuggingFace: gli elementi $-\infty$ impediscono il flusso di gradiente tra testo e codice durante la codifica, rispettando l'indipendenza richiesta dall'obiettivo JEPA.

C.2 Il predittore a pesi condivisi

Il predittore Pred(·) è implementato aggiungendo k ∈ {0,...,K} token speciali [PRED] alla fine della sequenza Text. L'embedding dell'ultimo token [PRED] all'ultimo strato del Transformer è l'output del predittore:

Pred(Enc(Text)) = h^{(last)}_{L+k}

Con k = 0, Pred(x) = x (predittore triviale, identità). Con k > 0, i token [PRED] elaborano l'informazione del contesto attraverso i meccanismi di auto-attenzione causale del Transformer, apprendendo trasformazioni non lineari senza parametri aggiuntivi rispetto al modello base.

C.3 Protocollo di selezione degli iperparametri

La selezione degli iperparametri avviene in due fasi. (1) Si identifica il learning rate ottimale lr ∈ {1e-5, 2e-5, 4e-5, 8e-5} per il baseline, scegliendo il valore che massimizza l'accuratezza dopo 4 epoche per la coppia (modello, dataset). (2) Mantenendo fisso il lr del baseline, si ricerca sulla griglia (k, λ) ∈ {0,1,2,3,4} × {0.5,1,2,4} la configurazione ottimale per LLM-JEPA. Ogni configurazione viene eseguita 5 volte con i semi {82, 23, 37, 84, 4}.

C.4 Risultati di fine-tuning per quattro famiglie di modelli

I risultati su NL-RX-SYNTH mostrano miglioramenti consistenti per tutti i modelli testati. Llama-3.2-1B-Instruct: baseline 57.29% ± 5.32%, LLM-JEPA 71.46% ± 1.34%

(miglioramento +14.17 pp, p=1.0e-3, configurazione k=1, λ=1). Gemma-2-2b-it: baseline 33.65% ± 3.24%, LLM-JEPA 43.12% ± 2.61% (+9.47 pp, p=5.5e-3, k=4, λ=2). OpenELM-1_1B-Instruct: baseline 12.07% ± 1.81%, LLM-JEPA 25.40% ± 2.40% (+13.33 pp, p=5.1e-4, k=3, λ=4). OLMo-2-0425-1B-Instruct: baseline 87.09% ± 0.36%, LLM-JEPA 87.52% ± 0.29% (+0.43 pp, p=2.5e-3, k=0, λ=2).

C.5 Risultati per dimensioni del modello crescenti

L'analisi della scalabilità mostra miglioramenti statisticamente significativi a tutte le scale. Llama-3.2-3B-Instruct: baseline 74.55% ± 3.58%, LLM-JEPA 77.16% ± 3.66% (p=0.0352, k=0, λ=2). Llama-3.1-8B-Instruct: baseline 35.77% ± 6.60%, LLM-JEPA 63.57% ± 16.81% (p=0.0131, k=0, λ=2.0). OLMo-2-1124-7B-Instruct: baseline 87.26% ± 0.27%, LLM-JEPA 87.75% ± 0.33% (p=0.0345, k=2, λ=20).

C.6 Fine-tuning LoRA

LLM-JEPA migliora sistematicamente rispetto al baseline a tutti i rank LoRA testati su NL-RX-SYNTH con Llama-3.2-1B-Instruct (lr=2e-5, k=1, λ=1):

Rank 32: baseline 6.09% ± 0.55%, LLM-JEPA 7.45% ± 1.87%.

Rank 64: baseline 21.09% ± 1.90%, LLM-JEPA 32.46% ± 1.26%.

Rank 128: baseline 34.21% ± 2.82%, LLM-JEPA 48.45% ± 3.66%.

Rank 256: baseline 45.57% ± 4.52%, LLM-JEPA 60.80% ± 2.31%.

Rank 512: baseline 50.18% ± 5.15%, LLM-JEPA 72.41% ± 2.94%.

Full fine-tuning: baseline 57.29% ± 5.32%, LLM-JEPA 71.46% ± 1.34%.

Crucialmente, con rank 512 (22.59% dei parametri addestrabili), LLM-JEPA raggiunge un'accuratezza comparabile al full fine-tuning (72.41% vs 71.46%), mentre il baseline LoRA rimane significativamente inferiore (50.18% vs 57.29%).

C.7 Analisi strutturale delle rappresentazioni

La qualità delle rappresentazioni indotte da LLM-JEPA è documentata attraverso l'analisi della matrice differenza Enc(Text) − Enc(Code). La norma dei valori singolari decresce di circa tre ordini di grandezza rispetto al modello base e al fine-tuning standard: i valori si trovano nell'ordine di 10^1 per LLM-JEPA contro 10^2 - 10^3 per gli approcci standard.

Questo è confermato dall'errore di regressione ai minimi quadrati $\min_X \|\text{Enc(Text)} \cdot X - \text{Enc(Code)}\|^2$: pari a 4.47 per LLM-JEPA k=1 e 4.04 per k=0, contro 3035 per il fine-tuning standard e 3953 per il modello base. La quasi-linearità della mappa tra rappresentazioni delle due viste è la manifestazione geometrica del miglioramento delle prestazioni.

C.8 Loss dropout e analisi computazionale

Con tasso di dropout LD = α, il costo per epoca è $(2 - \alpha) \cdot C_{base}$. I risultati sperimentali mostrano che LLM-JEPA con LD = 0.5, λ = 1 raggiunge 64.08% a 14.49 PFLOPs, superiore al caso senza dropout (55.60% a parità di costo). Con LD = 0.75, λ = 4 e 28.98 PFLOPs:

72.42%, il valore più alto fra tutte le configurazioni. La regola empirica per la co-regolazione degli iperparametri è $\lambda \cdot (1 - \alpha) \approx$ costante.

Appendice D — Studio di Ablazione di Var-JEPA e Diagnostici Distributivi

La presente appendice riporta i risultati completi dello studio di ablazione condotto da Gögl e Yau (2026) su un dataset sintetico controllato, con diagnostici distributivi per i latenti di contesto s_x e target s_y.

D.1 Setup della simulazione

I dati sintetici sono generati da un processo latente con: z_i ~ $\mathcal{N}$(0, Σ_z), s_{x_i} ~ ½$\mathcal{N}$(0, Σ_x) + ½$\mathcal{N}$(δ, Σ_x), s_{y_i} | s_{x_i}, z_i ~ $\mathcal{N}$(s_{x_i} + Az_i, Σ_y), x_i = h_x(s_{x_i}) + ε_x, y_i = h_y(s_{y_i}) + ε_y. I parametri sono: D_obs = 32, d_s = 16, d_z = 8, σ_x = σ_z = 1.0, σ_y = 0.5, δ_scale = 2.0, τ_x = τ_y = 0.3. Le mappe h_x e h_y sono MLP a due strati con attivazione tanh, fissate per l'intera esecuzione.

D.2 Varianti sperimentali

Si confrontano dieci varianti dell'obiettivo di addestramento, etichettate A–J. I pesi di default sono α^{rec} = α^{gen} = α^{KL}_{sx} = α^{KL}_z = α^{KL}_{sy} = 1 per i termini ELBO, e λ_{sx} = λ_{sy} = 10 per SIGReg.

(A) ELBO completo — Var-JEPA standard. (B) ELBO + SIGReg applicata solo a s_x (λ_{sy}=0). (C) ELBO + SIGReg applicata solo a s_y (λ_{sx}=0). (D) ELBO + SIGReg su entrambi i latenti. (E) ELBO con α^{KL}_{sx}=0 (KL su s_x rimossa). (F) ELBO con α^{KL}_{sy}=0 (KL su s_y rimossa). (G) ELBO con α^{rec}=α^{gen}=0 (solo KL, nessuna ricostruzione). (H) Come G ma con SIGReg aggiunta. (I) ELBO con tutti i KL rimossi (solo ricostruzione e generazione). (J) Come I ma con SIGReg aggiunta.

D.3 Risultati — Latente di contesto s_x

I diagnostici per la distribuzione aggregata q_φ(s_x) mostrano (media ± std su 5 run):

(A) ELBO: Acc=0.996±0.002, KL_agg=0.113±0.037, SIGReg-MSE=4.0e-4±1.0e-4, ‖Cov−I‖_F=0.649±0.127, $\|\mathbb{E}[s]\|_2$=0.082±0.009.

(B) ELBO+SIGReg(sx): Acc=0.996±0.002, KL_agg=0.108±0.032 — SIGReg su s_x non migliora ulteriormente rispetto a (A).

(E) ELBO senza KL_{sx}: Acc=0.996±0.002, KL_agg=8.374±1.944, SIGReg-MSE=5.1e-2 — collasso distributivo severo ma accuratezza mantenuta.

(G) Solo KL, nessuna ricostruzione: Acc=0.571±0.010, KL_agg=0.015 — accuratezza prossima alla casualità nonostante ottima distribuzione.

(I) Solo ricostruzione, nessuna KL: Acc=0.995±0.002, KL_agg=10.465±2.207 — rappresentazioni informative ma fortemente non-gaussiane.

Conclusione chiave: i termini KL per-sample e quelli di ricostruzione sono mutuamente necessari. Né l'uno né l'altro da solo produce rappresentazioni di qualità ottimale sia per l'accuratezza sia per la struttura distributiva.

D.4 Risultati — Latente target s_y

I diagnostici per s_y riflettono la struttura diversa del KL verso il prior condizionato appreso piuttosto che verso $\mathcal{N}(0,I)$:

(A) ELBO: Acc(s_y)=0.993±0.001, KL_agg(s_y)=3.530±0.377 — la deviazione dall'isotropia gaussiana è attesa e teoricamente corretta.

(C) ELBO+SIGReg(sy): Acc=0.993±0.002, KL_agg(s_y)=1.999±0.058 — SIGReg riduce la non-gaussianità aggregata di s_y senza degradare l'accuratezza.

(F) ELBO senza KL_{sy}: Acc=0.983±0.003, KL_agg(s_y)=6.201 — rimozione del KL di predizione degrada le prestazioni.

(G) Solo KL: Acc(s_y)=0.543±0.033 — collasso simmetrico per s_y.

Conclusione: la non-gaussianità aggregata di s_y (KL_agg ≈ 3.5 per la variante A) è una caratteristica teoricamente corretta, non un difetto: s_y è governato dal prior condizionato appreso p_θ(s_y|s_x,z), non da un prior fisso gaussiano standard.

D.5 Generazione degli embedding per la valutazione downstream

Per la valutazione downstream, Var-T-JEPA usa le medie posteriori come embedding deterministici: s_x = μ^{sx}_φ, z = μ^z_φ, s_w = μ^{sw}_φ. L'ausiliario z è inferito tramite l'auxiliary encoder q_φ(z|s_x) anziché campionato dal prior p(z), mantenendo coerenza con il regime di addestramento. Le stime di incertezza per la valutazione selettiva sono calcolate aggregando le deviazioni standard del posteriore target: media per dataset reali (AD, CO, EL, CC, BM), 90° percentile per dataset semi-sintetici (MNIST, SIM) per catturare i comportamenti di coda.

Bibliografia

La seguente bibliografia raccoglie i riferimenti citati nei capitoli della monografia, organizzati per autore in ordine alfabetico. Le opere dei tre contributi principali — Gögl e Yau (2026), Huang, LeCun e Balestriero (2025) e Mumuni e Mumuni (2025) — sono integrate con le loro rispettive bibliografie, selezionando i riferimenti più pertinenti alla trattazione.

Ajzen, I. e Madden, T. J. (1986). Prediction of goal-directed behavior: Attitudes, intentions, and perceived behavioral control. Journal of Experimental Social Psychology, 22(5), 453–474.

Assran, M., Duval, Q., Misra, I., Bojanowski, P., Vincent, P., Rabbat, M., LeCun, Y. e Ballas, N. (2023). Self-supervised learning from images with a joint-embedding predictive architecture. CVPR, pp. 15619–15629.

Baddeley, A. (1992). Working memory. Science, 255(5044), 556–559.

Balestriero, R. e LeCun, Y. (2024). Learning by reconstruction produces uninformative features for perception. arXiv:2402.11337.

Balestriero, R. e LeCun, Y. (2025). LeJEPA: Provable and scalable self-supervised learning without the heuristics.

Bardes, A., Garrido, Q., Ponce, J., Chen, X., Rabbat, M., LeCun, Y., Assran, M. e Ballas, N. (2024). Revisiting feature prediction for learning visual representations from video. arXiv:2404.08471.

Barrault, L., Duquenne, P.-A., Elbayad, M., et al. (2024). Large concept models: Language modeling in a sentence representation space. arXiv:2412.08821.

Blackard, J. (1998). Covertype. UCI Machine Learning Repository. doi:10.24432/C50K5N.

Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R. e Bengio, S. (2016). Generating sentences from a continuous space. CoNLL 2016, pp. 10–21.

Brown, T., Mann, B., Ryder, N., et al. (2020). Language models are few-shot learners. NeurIPS, 33, 1877–1901.

Chen, T. e Guestrin, C. (2016). XGBoost: A scalable tree boosting system. KDD '16, pp. 785–794.

Chen, T., Kornblith, S., Norouzi, M. e Hinton, G. (2020). A simple framework for contrastive learning of visual representations. ICML, pp. 1597–1607.

Clark, A. (2012). Magic words: How language augments human computation. In Language and meaning in cognitive science. Routledge, pp. 21–39.

Cobbe, K., Kosaraju, V., Bavarian, M., et al. (2021). Training verifiers to solve math word problems. arXiv:2110.14168.

DeepSeek-AI, Guo, D., Yang, D., et al. (2025). DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. arXiv:2501.12948.

Drozdov, K., Shwartz-Ziv, R. e LeCun, Y. (2024). Video representation learning with joint-embedding predictive architectures.

Epps, T. W. e Pulley, L. B. (1983). A test for normality based on the empirical characteristic function. Biometrika, 70(3), 723–726.

Gao, T., Yao, X. e Chen, D. (2021). SimCSE: Simple contrastive learning of sentence embeddings. arXiv:2104.08821.

Goertzel, B. (2014). Artificial general intelligence: Concept, state of the art, and future prospects. Journal of Artificial General Intelligence, 5(1), 1.

Goodfellow, I. J., Mirza, M., Xiao, D., Courville, A. e Bengio, Y. (2013). An empirical investigation of catastrophic forgetting in gradient-based neural networks. arXiv:1312.6211.

Gorishniy, Y., Rubachev, I., Khrulkov, V. e Babenko, A. (2021). Revisiting deep learning models for tabular data. NeurIPS, 34, 18932–18943.

Gögl, M. e Yau, C. (2026). Var-JEPA: A variational formulation of the joint-embedding predictive architecture — Bridging predictive and generative self-supervised learning. arXiv:2603.20111.

Grattafiori, A., Dubey, A., Jauhri, A., et al. (2024). The Llama 3 herd of models. arXiv:2407.21783.

Grill, J.-B., Strub, F., Altché, F., et al. (2020). Bootstrap your own latent: A new approach to self-supervised learning. NeurIPS, 33, 21271–21284.

Ha, D. e Schmidhuber, J. (2018). World models. arXiv:1803.10122.

Hao, X. e Shafto, P. (2023). Coupled variational autoencoder. ICML 2023, pp. 12546–12555.

Harnad, S. (1990). The symbol grounding problem. Physica D: Nonlinear Phenomena, 42(1-3), 335–346.

Harries, M. (1999). Splice-2 comparative evaluation: Electricity pricing. Technical report, UNSW.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P. e Girshick, R. (2022). Masked autoencoders are scalable vision learners. CVPR, pp. 16000–16009.

He, K., Zhang, X., Ren, S. e Sun, J. (2016). Deep residual learning for image recognition. CVPR, pp. 770–778.

Higgins, I., Matthey, L., Pal, A., et al. (2017). β-VAE: Learning basic visual concepts with a constrained variational framework. ICLR.

Hoffman, M. D. e Johnson, M. J. (2016). ELBO surgery: Yet another way to carve up the variational evidence lower bound. NeurIPS 2016 Workshop.

Huang, H., LeCun, Y. e Balestriero, R. (2025). LLM-JEPA: Large language models meet joint embedding predictive architectures. arXiv:2509.14252.

Huang, Y. (2026). VJEPA: Variational joint embedding predictive architectures as probabilistic world models.

Jimenez, C. E., Yang, J., Wettig, A., et al. (2024). SWE-bench: Can language models resolve real-world GitHub issues? ICLR 2024.

Jing, L., Vincent, P., LeCun, Y. e Tian, Y. (2021). Understanding dimensional collapse in contrastive self-supervised learning. arXiv:2110.09348.

Kenneweg, T., Kenneweg, P. e Hammer, B. (2025). JEPA for RL: Investigating joint-embedding predictive architectures for reinforcement learning. arXiv:2504.16591.

Kingma, D. P. e Welling, M. (2013). Auto-encoding variational Bayes. arXiv:1312.6114.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., et al. (2017). Overcoming catastrophic forgetting in neural networks. PNAS, 114(13), 3521–3526.

Kohavi, R. (1996). Scaling up the accuracy of naive-Bayes classifiers: A decision-tree hybrid. AAAI Press.

LeCun, Y. (2022). A path towards autonomous machine intelligence (version 0.9.2). OpenReview.

LeCun, Y., Bottou, L., Bengio, Y. e Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324.

Lee, K., Chang, M.-W. e Toutanova, K. (2019). Latent retrieval for weakly supervised open domain question answering. ACL 2019, pp. 6086–6096.

Lewis, P., Perez, E., Piktus, A., et al. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. NeurIPS, 33, 9459–9474.

Littwin, E., Saremi, O., Advani, M., et al. (2024). How JEPA avoids noisy features: The implicit bias of deep linear self distillation networks. NeurIPS, 37, 91300–91336.

Locascio, N., Narasimhan, K., DeLeon, E., Kushman, N. e Barzilay, R. (2016). Neural generation of regular expressions from natural language with minimal domain knowledge. EMNLP 2016, pp. 1918–1923.

Lupyan, G. (2016). The centrality of language in human cognition. Language Learning, 66(3), 516–553.

Mehta, S., Sekhavat, M. H., Cao, Q., et al. (2024). OpenELM: An efficient language model family with open training and inference framework. arXiv:2404.14619.

Moro, S., Rita, P. e Cortez, P. (2014). Bank Marketing. UCI Machine Learning Repository. doi:10.24432/C5K306.

Mumuni, A. e Mumuni, F. (2025). Large language models for artificial general intelligence (AGI): A survey of foundational principles and approaches. arXiv:2501.03151.

Murphy, K. P. (2022). Probabilistic Machine Learning: An introduction. MIT Press.

Newell, A. e Simon, H. A. (1975). Computer science as empirical inquiry: Symbols and search. ACM Turing Award Lectures.

Ni, J., Hernandez Abrego, G., Constant, N., et al. (2021). Sentence-T5: Scalable sentence encoders from pre-trained text-to-text models. arXiv:2108.08877.

OLMo, P. W., Soldaini, L., Groeneveld, D., et al. (2024). 2 OLMo 2 furious. arXiv:2501.00656.

OpenAI (2023). GPT-4 technical report. arXiv:2303.08774.

Ouyang, L., Wu, J., Jiang, X., et al. (2022). Training language models to follow instructions with human feedback. NeurIPS, 35, 27730–27744.

Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, L., Shazeer, N., Ku, A. e Tran, D. (2018). Image Transformer. ICML, pp. 4052–4061.

Pearl, J. (2009). Causality. Cambridge University Press.

Pearl, J. (2019). The seven tools of causal inference, with reflections on machine learning. Communications of the ACM, 62(3), 54–60.

Radford, A., Kim, J. W., Hallacy, C., et al. (2021). Learning transferable visual models from natural language supervision. ICML, pp. 8748–8763.

Raffel, C., Shazeer, N., Roberts, A., et al. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. JMLR.

Reimers, N. e Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using siamese BERT-networks. arXiv:1908.10084.

Song, W., Shi, C., Xiao, Z., et al. (2019). AutoInt: Automatic feature interaction learning via self-attentive neural networks. CIKM '19, pp. 1161–1170.

Team, G., Riviere, M., Pathak, S., et al. (2024). Gemma 2: Improving open language models at a practical size. arXiv:2408.00118.

Thimonier, H., Costa, J. L. D. M., Popineau, F., Rimmel, A. e Doan, B.-L. T. (2025). T-JEPA: Augmentation-free self-supervised learning for tabular data. ICLR 2025.

van den Oord, A., Kalchbrenner, N. e Kavukcuoglu, K. (2016). Pixel recurrent neural networks. ICML, pp. 1747–1756.

van den Oord, A., Li, Y. e Vinyals, O. (2018). Representation learning with contrastive predictive coding. arXiv:1807.03748.

Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is all you need. NeurIPS, 30.

Wang, R., Shivanna, R., Cheng, D., et al. (2021). DCN V2: Improved deep & cross network and practical lessons for web-scale learning to rank systems. WWW '21, pp. 1785–1797.

Wang, B. e Sun, H. (2025). Is the reversal curse a binding problem? Uncovering limitations of transformers from a basic generalization failure. arXiv:2504.01928.

Wang, L., Yang, N., Huang, X., et al. (2022). Text embeddings by weakly-supervised contrastive pre-training. arXiv:2212.03533.

Wei, J., Bosma, M., Zhao, V., et al. (2022). Finetuned language models are zero-shot learners. ICLR 2022.

Yang, A., Li, A., Yang, B., et al. (2025). Qwen3 technical report. arXiv:2505.09388.

Ye, X., Chen, Q., Wang, X., Dillig, I. e Durrett, G. (2020). Sketch-driven regular expression generation from natural language and examples. TACL, 8, 679–694.

Yeh, I.-C. (2009). Default of Credit Card Clients. UCI Machine Learning Repository. doi:10.24432/C55S3H.

Yu, T., Zhang, R., Yang, K., et al. (2018). Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. EMNLP 2018.

Zečević, M., Willig, M., Dhami, D. S. e Kersting, K. (2023). Causal parrots: Large language models may talk causality but are not causal. arXiv:2308.13067.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A. e Choi, Y. (2019). HellaSwag: Can a machine really finish your sentence? ACL 2019, pp. 4791–4800.

Zhong, Z., Guo, J., Yang, W., et al. (2018). SemRegex: A semantics-based approach for generating regular expressions from natural language specifications. EMNLP 2018.

Ziegler, D. M., Stiennon, N., Wu, J., et al. (2019). Fine-tuning language models from human preferences. arXiv:1909.08593.